TRABAJO DE GRADO
Proyecto de Grado

Diseño y Evaluación de un Sistema de Identificación Facial de Bajo Costo

Corporación Universitaria Remington.
Facultad de Ingenierías.
Programa de Ingeniería de Sistemas.

Santiago Quitian Rodríguez.
Director: Jonatan Stick Campos.
Opción en la que realizó su trabajo de grado (Proyecto de Grado).
2025.

Dedicatoria

Dedico este trabajo de grado al profesor **Jonatan Stick Campos**, cuya orientación, paciencia y confianza inquebrantable fueron el motor que impulsó cada etapa de esta investigación. Su guía académica, su disposición constante para compartir su conocimiento y, sobre todo, su capacidad para creer en este proyecto cuando aún era solo una idea, hizo posible transformar un desafío técnico en una realidad concreta. Gracias por enseñarme que la verdadera ingeniería no solo resuelve problemas, sino que democratiza soluciones.

Tabla de Contenidos

Lista de tablas

(Incluya esta sección sólo si aplica para su trabajo de grado)

Lista de figuras

(Incluya esta sección sólo si aplica para su trabajo de grado)

Resumen

Las grandes superficies comerciales e instituciones públicas enfrentan desafíos crecientes en materia de seguridad y control de acceso. Los métodos tradicionales, como tarjetas magnéticas, sistemas de identificación manual o terminales de acceso cerrados, presentan vulnerabilidades significativas, limitaciones de escalabilidad y deficiencias en la gestión centralizada. Si bien el reconocimiento facial ha emergido como una alternativa tecnológica moderna y eficiente, su adopción se ve limitada por los altos costos de licenciamiento de los Sistemas de Gestión de Video Empresarial (VMS) comerciales, que pueden superar los $400.000 COP por cámara, representando una barrera económica significativa para pequeñas y medianas organizaciones.

Esta investigación aborda dicha brecha económica mediante el diseño y evaluación de un sistema de identificación facial de bajo costo, orientado a su implementación en grandes superficies, garantizando eficiencia técnica, accesibilidad económica y viabilidad de despliegue. El proyecto se desarrolló bajo un enfoque de investigación aplicada con metodología cuantitativa en entorno controlado, integrando herramientas de software libre (OpenCV, algoritmos Haar Cascade, EigenFace, FisherFace y LBPH) con hardware accesible (cámaras USB, IP de múltiples fabricantes y dispositivos Raspberry Pi 4 y 5).

El sistema implementado demostró compatibilidad con múltiples tipos de cámaras mediante protocolo RTSP, capacidad de procesamiento en tiempo real sin servidores externos, y escalabilidad comprobada al gestionar exitosamente hasta 4 flujos de video simultáneos en Raspberry Pi 5. Las pruebas de validación arrojaron una precisión de detección del 92% utilizando el dataset Labeled Faces in the Wild (LFW), confirmando la

eficacia del enfoque algorítmico en plataformas de procesamiento de bajo costo. El análisis económico comparativo evidenció una reducción de costos superior al 85% frente a soluciones VMS comerciales equivalentes.

Los resultados demuestran que es técnica y económicamente viable desarrollar soluciones de identificación facial funcionales, precisas y escalables sin recurrir a tecnologías propietarias costosas. Este enfoque promueve la democratización tecnológica con potencial de aplicación en instituciones educativas, pequeñas y medianas empresas, y sectores públicos que requieren sistemas de seguridad y control de acceso económicamente accesibles.

Palabras clave

identificación facial, bajo costo, OpenCV, Raspberry Pi, visión artificial, RTSP, VMS de código abierto

Introducción, Marco teórico o de referencia

La seguridad y el control de acceso representan desafíos fundamentales para las organizaciones contemporáneas, especialmente en entornos de alta circulación como centros comerciales, instituciones educativas, complejos empresariales y edificios gubernamentales. Durante décadas, estos espacios han dependido de métodos convencionales de identificación que, si bien ampliamente adoptados, presentan limitaciones inherentes que comprometen tanto la eficiencia operativa como la seguridad real de las instalaciones.

Los sistemas tradicionales basados en tarjetas de proximidad, códigos PIN o registros manuales enfrentan vulnerabilidades documentadas, las credenciales pueden ser extraviadas, duplicadas o transferidas entre usuarios no autorizados, comprometiendo la integridad del sistema de acceso. Además, estos métodos requieren inversiones continuas en la emisión y reemplazo de tarjetas, gestión de bases de datos de credenciales, y personal dedicado a la supervisión de accesos, generando costos operativos recurrentes que impactan significativamente los presupuestos institucionales.

En respuesta a estas limitaciones, la identificación biométrica ha emergido como una alternativa tecnológica que ofrece mayor seguridad al vincular directamente la identidad de una persona con características físicas únicas e intransferibles. Entre las diversas modalidades biométricas disponibles, el reconocimiento facial ha ganado particular relevancia debido a su naturaleza no invasiva, su capacidad de operar a distancia

sin requerir contacto físico o cooperación activa del usuario, y su compatibilidad con infraestructuras de videovigilancia existentes.

Sin embargo, la implementación de sistemas de reconocimiento facial en el contexto latinoamericano, y particularmente en Colombia, enfrenta una barrera económica considerable. Los Sistemas de Gestión de Video Empresarial (VMS) que integran capacidades de identificación facial comercializados por fabricantes establecidos como Milestone, Genetec, Avigilon o las propias soluciones propietarias de fabricantes de cámaras como Hikvision, Dahua y Hanwha, imponen estructuras de licenciamiento que pueden superar los $400.000 COP por cámara. Para una instalación de mediana escala con 20 cámaras, esto representa una inversión superior a los $8.000.000 COP únicamente en licencias de software, sin considerar los costos de hardware, instalación, capacitación y mantenimiento.

Esta realidad económica excluye efectivamente a pequeñas y medianas empresas, instituciones educativas públicas, y organizaciones sin fines de lucro del acceso a tecnologías de seguridad avanzadas, perpetuando una brecha tecnológica donde solo las grandes corporaciones pueden implementar sistemas de identificación facial funcionales y escalables. Los dispositivos de identificación facial de bajo costo disponibles en el mercado, como los terminales independientes fabricados por empresas como ZKTeco, ofrecen soluciones económicas, pero carecen de las capacidades de gestión centralizada, escalabilidad e integración con infraestructuras de video existentes que requieren las instalaciones de mediana y gran escala.

El presente trabajo de investigación aborda esta problemática mediante el diseño, implementación y evaluación de un sistema de identificación facial basado completamente en tecnologías de código abierto y hardware de bajo costo. Utilizando la biblioteca OpenCV para procesamiento de visión por computadora, algoritmos de reconocimiento facial de dominio público, el protocolo RTSP para streaming de video estándar, y dispositivos de computación de propósito general como Raspberry Pi, esta investigación demuestra la viabilidad técnica y económica de desarrollar soluciones funcionales sin depender de licencias comerciales propietarias.

La relevancia de este trabajo trasciende el ámbito puramente tecnológico para insertarse en una discusión más amplia sobre democratización del acceso a tecnologías de seguridad avanzadas. Al documentar un proceso replicable de desarrollo e implementación que reduce los costos en entre un 70 y 85 % comparado con soluciones comerciales equivalentes (véase en la página 79: **Análisis de Viabilidad Económica**), esta investigación contribuye al conocimiento técnico disponible para instituciones con recursos limitados que buscan mejorar sus capacidades de seguridad sin comprometer su sostenibilidad financiera.

El documento se estructura de la siguiente manera: primero se presenta un marco teórico que contextualiza los fundamentos de reconocimiento facial, los algoritmos implementados, y el estado del arte en sistemas VMS de código abierto. Posteriormente se detalla la metodología empleada para el diseño e implementación del sistema, seguida de la presentación y análisis de resultados obtenidos en pruebas de funcionalidad, precisión y escalabilidad. Finalmente, se discuten las conclusiones, limitaciones identificadas y

posibles líneas de trabajo futuro que puedan continuar expandiendo las capacidades del sistema desarrollado.

1.1. Sistemas Biométricos y Reconocimiento Facial

Los sistemas biométricos constituyen tecnologías de identificación y verificación de identidad basadas en características físicas o comportamentales únicas e inherentes a cada individuo. A diferencia de los métodos tradicionales de autenticación que dependen de elementos externos como contraseñas, tarjetas o tokens, la biometría utiliza atributos biológicos que son intransferibles, difíciles de falsificar y permanecen relativamente estables a lo largo del tiempo.

Jain, Ross y Prabhakar (2004) definen un sistema biométrico como un sistema de reconocimiento de patrones que establece la autenticidad de una característica fisiológica o de comportamiento específica poseída por un usuario. Entre las modalidades biométricas más comunes se encuentran las huellas dactilares, el reconocimiento de iris, el escaneo de retina, el reconocimiento de voz, la geometría de la mano y el reconocimiento facial.

El reconocimiento facial se distingue de otras modalidades biométricas por varias características que lo hacen particularmente atractivo para aplicaciones de control de acceso en espacios públicos. En primer lugar, es un método no invasivo que no requiere contacto físico directo con dispositivos de captura, aspecto que ha cobrado mayor relevancia en contextos post-pandémicos donde se priorizan soluciones sin contacto. En

segundo lugar, puede operar a distancia y sin la cooperación activa del usuario, permitiendo la identificación de personas en movimiento o en situaciones donde la interacción directa con un sensor no es práctica o deseable.

Zhao et al. (2003) clasifican los sistemas de reconocimiento facial en dos categorías fundamentales según su propósito operacional. Los sistemas de verificación facial, también conocidos como autenticación uno-a-uno, responden a la pregunta: ¿es esta persona quien dice ser? En este escenario, el sistema compara la imagen capturada con una plantilla específica asociada a una identidad declarada, generando una decisión binaria de aceptación o rechazo. Por otro lado, los sistemas de identificación facial, o búsqueda uno-a-muchos, abordan la pregunta: ¿quién es esta persona? Aquí el sistema compara la imagen capturada contra una base de datos completa de identidades conocidas, determinando la identidad más probable o declarando que la persona es desconocida si no se encuentra coincidencia suficiente.

El proceso de reconocimiento facial generalmente comprende cuatro etapas fundamentales. La primera etapa corresponde a la detección facial, donde el sistema localiza y delimita la región del rostro dentro de una imagen que puede contener múltiples objetos, fondos complejos y variaciones de iluminación. La segunda etapa implica el alineamiento facial, proceso mediante el cual se normalizan las variaciones de pose, escala y orientación para facilitar la comparación posterior. La tercera etapa consiste en la extracción de características, donde se identifican y codifican los rasgos distintivos del

rostro en una representación matemática compacta. Finalmente, la cuarta etapa realiza la comparación de estas características contra las plantillas almacenadas en la base de datos, utilizando métricas de similitud para determinar la identidad.

A pesar de sus ventajas, el reconocimiento facial enfrenta desafíos técnicos significativos que han sido objeto de investigación continua. Las variaciones en iluminación pueden alterar drásticamente la apariencia facial en las imágenes capturadas. Los cambios de pose, donde el rostro no está directamente orientado hacia la cámara, introducen distorsiones geométricas que dificultan la comparación. Las expresiones faciales modifican temporalmente la configuración de los rasgos faciales. El envejecimiento produce cambios graduales en la apariencia facial a lo largo del tiempo. Accesorios como gafas, gorras o mascarillas pueden ocluir parcialmente el rostro. La resolución de las imágenes capturadas afecta directamente la cantidad de información facial disponible para el análisis.

En el contexto de esta investigación, el sistema desarrollado implementa capacidades de identificación facial uno-a-muchos, diseñado específicamente para escenarios de control de acceso donde se busca determinar si una persona pertenece a un conjunto predefinido de usuarios autorizados, sin requerir que el individuo declare su identidad previamente.

1.2. Fundamentos de Visión por Computadora

La visión por computadora es una disciplina de la inteligencia artificial que permite a los sistemas informáticos interpretar y comprender información visual del mundo real. Szeliski (2010) la define como el proceso de transformar datos de imágenes en descripciones del mundo que tienen sentido para un propósito específico y pueden producir acciones apropiadas. Esta capacidad de extraer información significativa de imágenes digitales ha revolucionado múltiples campos, desde la manufactura automatizada hasta los sistemas de seguridad y vigilancia.

El procesamiento digital de imágenes constituye el fundamento técnico sobre el cual se construyen los sistemas de visión por computadora. Una imagen digital se representa matemáticamente como una matriz bidimensional de píxeles, donde cada píxel contiene información sobre la intensidad lumínica o los valores de color en una ubicación específica. En imágenes en escala de grises, cada píxel se representa mediante un valor numérico que típicamente varía entre 0 (negro absoluto) y 255 (blanco absoluto). En imágenes a color, cada píxel contiene tres valores correspondientes a los canales rojo, verde y azul del espacio de color RGB.

Las operaciones fundamentales del procesamiento de imágenes incluyen transformaciones que modifican los valores de los píxeles para mejorar características relevantes o suprimir información irrelevante. El suavizado de imágenes mediante filtros gaussianos reduce el ruido aleatorio que puede interferir con algoritmos de detección

posteriores. La detección de bordes identifica discontinuidades bruscas en la intensidad lumínica que frecuentemente corresponden a límites entre objetos. El ajuste de histogramas normaliza la distribución de intensidades para mejorar el contraste en condiciones de iluminación subóptimas. La segmentación divide la imagen en regiones significativas que pueden procesarse independientemente.

En el contexto específico del reconocimiento facial, la visión por computadora enfrenta el desafío de extraer representaciones invariantes del rostro humano que permanezcan estables ante transformaciones geométricas, variaciones fotométricas y oclusiones parciales. Turk y Pentland (1991) demostraron que los rostros humanos ocupan un subespacio de menor dimensión dentro del espacio completo de todas las imágenes posibles, y que este subespacio puede caracterizarse mediante técnicas de reducción de dimensionalidad como el Análisis de Componentes Principales.

La detección facial, como etapa preliminar del reconocimiento, requiere identificar la presencia y ubicación de rostros en imágenes donde pueden coexistir con múltiples objetos de fondo. Los algoritmos de detección facial deben ser robustos ante variaciones de escala, dado que los rostros pueden aparecer a diferentes distancias de la cámara, rotaciones en el plano de la imagen, y variaciones de iluminación que modifican la apariencia visual del rostro sin alterar su identidad.

Viola y Jones (2004) revolucionaron la detección facial en tiempo real mediante la introducción de características Haar, que son patrones rectangulares simples que capturan diferencias de intensidad entre regiones adyacentes de la imagen. Estas características, combinadas con un algoritmo de aprendizaje en cascada, permiten evaluar rápidamente miles de ubicaciones potenciales en la imagen, descartando tempranamente regiones que claramente no contienen rostros y dedicando mayor procesamiento únicamente a regiones prometedoras.

La extracción de características faciales posteriores a la detección busca codificar la información discriminativa del rostro en una representación compacta que facilite la comparación entre diferentes instancias. Las características geométricas miden distancias y ángulos entre puntos de referencia faciales como los ojos, la nariz y la boca. Las características basadas en apariencia analizan la textura y distribución de intensidades en regiones específicas del rostro. Las características basadas en aprendizaje profundo emplean redes neuronales convolucionales para aprender automáticamente representaciones jerárquicas que capturan patrones complejos difícilmente codificables mediante reglas manuales.

1.3. Algoritmos de Reconocimiento Facial

El reconocimiento facial automático ha sido objeto de investigación intensiva durante las últimas cuatro décadas, resultando en el desarrollo de múltiples enfoques algorítmicos con diferentes fundamentos matemáticos y características operacionales. Esta

sección examina los algoritmos implementados en el sistema desarrollado, desde la detección inicial mediante clasificadores en cascada hasta los tres métodos de reconocimiento evaluados.

1.3.1. Haar Cascade para Detección Facial.

Los clasificadores en cascada basados en características Haar representan uno de los métodos más eficientes para la detección de objetos en tiempo real. Desarrollado originalmente por Viola y Jones (2001), este enfoque revolucionó el campo al proporcionar tasas de detección superiores al 95% con velocidades de procesamiento que permiten analizar múltiples cuadros de video por segundo en hardware convencional.

Las características Haar son representaciones simples que capturan patrones de contraste entre regiones adyacentes de la imagen. Una característica Haar típica consiste en dos o más rectángulos adyacentes, uno claro y otro oscuro, cuyo valor se calcula como la diferencia entre la suma de intensidades de píxeles en la región clara y la suma en la región oscura. Por ejemplo, una característica de dos rectángulos horizontales con la región superior clara y la inferior oscura captura eficazmente el patrón de contraste entre la frente y los ojos en un rostro frontal.

La ventaja computacional fundamental de las características Haar radica en su cálculo eficiente mediante imágenes integrales. Una imagen integral es una representación preprocesada donde cada posición contiene la suma acumulada de todos los píxeles

ubicados arriba y a la izquierda de esa posición. Con esta representación, el cálculo de la suma de píxeles en cualquier región rectangular requiere únicamente cuatro operaciones de acceso a memoria y tres sumas, independientemente del tamaño de la región.

El clasificador en cascada organiza múltiples etapas de clasificación secuenciales, cada una diseñada para rechazar rápidamente ventanas de imagen que claramente no contienen rostros, mientras permite que las ventanas prometedoras avancen a etapas posteriores más discriminativas. Las primeras etapas de la cascada emplean pocas características simples y umbrales permisivos, descartando aproximadamente el 50% de las ventanas negativas con procesamiento mínimo. Las etapas posteriores utilizan mayor número de características más complejas, aplicándose únicamente a la pequeña fracción de ventanas que superaron las etapas iniciales.

Este diseño en cascada resulta extremadamente eficiente en la práctica porque las imágenes típicas contienen muy pocas regiones que realmente corresponden a rostros. La mayoría de las ventanas de evaluación se rechazan en las primeras etapas con costo computacional mínimo, concentrando el procesamiento intensivo únicamente en las regiones más prometedoras.

En el sistema implementado, se utiliza el clasificador Haar Cascade preentrenado proporcionado por OpenCV, específicamente el modelo haarcascade_frontalface_default.xml, entrenado con miles de ejemplos positivos de rostros

frontales y negativos de no-rostros. Este clasificador detecta rostros en orientación frontal con variaciones de aproximadamente 45 grados en los ángulos de rotación, siendo menos efectivo para rostros de perfil o con inclinaciones pronunciadas.

1.3.2. EigenFaces (Análisis de Componentes Principales).

El método EigenFaces, introducido por Turk y Pentland (1991), representa uno de los enfoques fundacionales en reconocimiento facial automático. Se basa en la observación de que, aunque las imágenes faciales existen en un espacio de muy alta dimensionalidad (por ejemplo, una imagen de 100×100 píxeles tiene 10,000 dimensiones), los rostros humanos reales ocupan un subespacio de dimensión mucho menor dentro de este espacio completo.

El Análisis de Componentes Principales (PCA, por sus siglas en inglés) es una técnica estadística de reducción de dimensionalidad que identifica las direcciones de máxima varianza en un conjunto de datos. Aplicado al reconocimiento facial, PCA identifica los "rostros propios" o eigenfaces, que son los vectores propios de la matriz de covarianza del conjunto de imágenes faciales de entrenamiento.

El proceso de construcción del espacio EigenFaces comienza con un conjunto de N imágenes faciales de entrenamiento, todas normalizadas a la misma resolución y convertidas en vectores unidimensionales mediante el apilamiento de las filas de píxeles. Se calcula el rostro promedio como la media aritmética de todos los vectores de

entrenamiento. Cada imagen de entrenamiento se centra restando este rostro promedio, produciendo vectores de diferencia que capturan las desviaciones de cada rostro respecto al promedio.

La matriz de covarianza de estos vectores centrados captura las correlaciones entre diferentes posiciones de píxeles a través del conjunto de entrenamiento. Los vectores propios de esta matriz de covarianza, ordenados por sus valores propios correspondientes, constituyen los eigenfaces. Los eigenfaces con mayores valores propios capturan los modos principales de variación en el conjunto de rostros, mientras que aquellos con valores propios pequeños representan variaciones menores que frecuentemente corresponden a ruido o detalles irrelevantes para la identificación.

Para reconocer un rostro desconocido, la imagen se proyecta sobre el subespacio definido por los k eigenfaces más significativos, donde k típicamente varía entre 50 y 200 dependiendo del tamaño del conjunto de entrenamiento. Esta proyección produce un vector de coeficientes de dimensión k que representa el rostro en el espacio reducido. La distancia euclidiana entre este vector y los vectores correspondientes a los rostros conocidos en la base de datos determina la identidad más probable.

Las ventajas principales de EigenFaces incluyen su simplicidad conceptual y computacional, su capacidad de reducir drásticamente la dimensionalidad manteniendo la mayor parte de la información discriminativa, y su fundamento matemático riguroso. Sin

embargo, presenta limitaciones significativas. Es sensible a variaciones de iluminación, ya que PCA no distingue entre variaciones causadas por identidad y variaciones causadas por condiciones de captura. Requiere alineamiento preciso de los rostros, dado que pequeñas traslaciones o rotaciones en el plano de la imagen modifican significativamente la representación en el espacio de píxeles. Trata todas las regiones de la imagen con igual importancia, sin priorizar características faciales discriminativas como ojos, nariz y boca.

1.3.3. FisherFaces (Análisis Discriminante Lineal).

El método FisherFaces, propuesto por Belhumeur, Hespanha y Kriegman (1997), aborda algunas de las limitaciones fundamentales de EigenFaces mediante la aplicación del Análisis Discriminante Lineal (LDA, por sus siglas en inglés). Mientras que PCA busca direcciones de máxima varianza sin considerar las etiquetas de clase, LDA busca explícitamente las proyecciones que maximizan la separación entre clases diferentes mientras minimizan la dispersión dentro de cada clase.

El fundamento teórico de LDA deriva del criterio de Fisher, que define la calidad de una proyección mediante la razón entre la dispersión entre clases y la dispersión dentro de las clases. Una buena proyección agrupa las muestras pertenecientes a la misma persona mientras separa claramente las muestras de personas diferentes. Matemáticamente, esto se formula como la maximización de la razón entre la matriz de dispersión entre clases y la matriz de dispersión dentro de las clases.

El proceso de construcción del espacio FisherFaces comienza aplicando PCA para reducir la dimensionalidad inicial a un espacio intermedio, típicamente reteniendo suficientes componentes para preservar el 95% de la varianza total. Esta reducción preliminar es necesaria porque la matriz de dispersión dentro de las clases puede ser singular en el espacio original de alta dimensionalidad, especialmente cuando el número de muestras de entrenamiento es menor que la dimensionalidad de las imágenes.

Sobre este espacio intermedio reducido por PCA, se aplica LDA para encontrar las direcciones que mejor discriminan entre las diferentes identidades. El número máximo de direcciones discriminantes es C-1, donde C es el número de clases (personas) en el conjunto de entrenamiento. Cada rostro desconocido se proyecta sobre este subespacio discriminante, y la clasificación se realiza mediante el clasificador de distancia mínima, asignando la identidad cuyo centroide en el espacio proyectado está más cercano al vector proyectado del rostro desconocido.

FisherFaces presenta ventajas significativas sobre EigenFaces en escenarios con variaciones de iluminación controladas. Al maximizar explícitamente la separabilidad entre clases, produce representaciones más robustas para discriminación de identidad. La proyección discriminante tiende a suprimir variaciones dentro de la clase que no contribuyen a la distinción entre personas diferentes. Sin embargo, requiere múltiples imágenes de entrenamiento por persona para estimar confiablemente las matrices de dispersión dentro y entre clases. Es más sensible al sobreajuste cuando el número de

muestras de entrenamiento es limitado. La complejidad computacional del entrenamiento es mayor que PCA debido al cálculo de múltiples matrices de dispersión.

1.3.4. LBPH (Local Binary Patterns Histograms).

El método LBPH (Local Binary Patterns Histograms) representa un enfoque fundamentalmente diferente al reconocimiento facial, basado en la descripción de texturas locales en lugar de transformaciones globales del espacio de características. Ahonen, Hadid y Pietikäinen (2006) demostraron que los patrones binarios locales proporcionan una representación robusta y discriminativa para el reconocimiento facial, con ventajas particulares en términos de invariancia a cambios de iluminación monótonos y eficiencia computacional.

El operador LBP (Local Binary Pattern) analiza el vecindario de cada píxel en la imagen, comparando su intensidad con la de los píxeles circundantes. En su forma más simple, para cada píxel central, se examinan los 8 píxeles vecinos en un patrón circular. Si la intensidad de un vecino es mayor o igual a la del píxel central, se asigna un valor binario de 1; de lo contrario, se asigna 0. Estos 8 valores binarios, leídos en orden circular, forman un número binario de 8 bits que constituye el código LBP de ese píxel.

El código LBP resultante captura información sobre la estructura de micropatrones en el vecindario local. Por ejemplo, un píxel ubicado en una región plana uniforme tendrá un código LBP de 0 o 255, mientras que un píxel en un borde tendrá un patrón que refleja

la transición de intensidad. Las esquinas, líneas y otras microestructuras producen códigos LBP característicos. Esta codificación es intrínsecamente robusta a cambios de iluminación monótonos, ya que la operación de comparación binaria preserva las relaciones de orden de intensidad incluso cuando los valores absolutos cambian.

Para el reconocimiento facial, la imagen del rostro se divide en una cuadrícula de regiones locales, típicamente 8×8 o 7×7 regiones rectangulares. Para cada región, se calcula el histograma de códigos LBP, que representa la distribución de frecuencia de los 256 patrones binarios posibles en esa área. Estos histogramas locales se concatenan en un único vector de características que describe el rostro completo, preservando tanto la información de textura local como la estructura espacial aproximada.

La comparación entre dos rostros se realiza mediante medidas de similitud de histogramas. La distancia Chi-cuadrado es particularmente efectiva para comparar distribuciones de frecuencia, calculando la suma ponderada de las diferencias al cuadrado entre bins correspondientes de los histogramas. Alternativamente, pueden emplearse métricas como la intersección de histogramas o la distancia euclidiana simple.

LBPH presenta varias ventajas distintivas que lo hacen especialmente atractivo para aplicaciones prácticas. Es robusto ante cambios de iluminación monótonos debido a la naturaleza relativa de las comparaciones binarias. No requiere alineamiento perfecto de los rostros, siendo tolerante a pequeñas traslaciones y deformaciones locales. Puede

actualizarse incrementalmente con nuevas imágenes sin reentrenar completamente el modelo. La eficiencia computacional es excelente tanto en entrenamiento como en reconocimiento. Funciona razonablemente bien incluso con conjuntos de entrenamiento pequeños.

Las limitaciones incluyen sensibilidad a cambios de iluminación no monótonos, como sombras duras que invierten las relaciones de orden de intensidad. Menor invariancia a rotaciones pronunciadas o cambios de escala significativos. La representación basada en histogramas descarta información sobre la ubicación exacta de los patrones dentro de cada región.

En el contexto del sistema desarrollado, LBPH se implementa con división de la imagen facial en regiones de 8×8, utilizando el operador LBP uniforme que considera únicamente los 59 patrones más comunes (aquellos con dos o menos transiciones de 0 a 1 en su representación binaria circular), reduciendo la dimensionalidad del histograma mientras preserva la información discriminativa más relevante.

1.4. Biblioteca OpenCV

OpenCV (Open Source Computer Vision Library) es una biblioteca de software de código abierto diseñada específicamente para aplicaciones de visión por computadora y aprendizaje automático. Inicialmente desarrollada por Intel en 1999 y actualmente mantenida por una comunidad global de desarrolladores, OpenCV se ha consolidado como

el estándar de facto para el desarrollo de aplicaciones de procesamiento de imágenes y video en tiempo real.

La biblioteca proporciona más de 2,500 algoritmos optimizados que abarcan un amplio espectro de tareas de visión por computadora. Estos incluyen detección y reconocimiento de objetos, seguimiento de movimiento, análisis de escenas, reconstrucción tridimensional, procesamiento de imágenes médicas, análisis de video de vigilancia, y reconocimiento de gestos, entre muchos otros. OpenCV está escrita principalmente en C++ para maximizar el rendimiento, pero proporciona interfaces para múltiples lenguajes de programación incluyendo Python, Java y MATLAB.

Una característica fundamental de OpenCV es su orientación hacia aplicaciones de tiempo real. Los algoritmos incluidos están altamente optimizados para eficiencia computacional, aprovechando cuando es posible las capacidades de procesamiento paralelo de los procesadores modernos mediante instrucciones SIMD (Single Instruction, Multiple Data). La biblioteca también puede configurarse para utilizar aceleración por GPU mediante CUDA en sistemas con tarjetas gráficas NVIDIA, multiplicando significativamente el rendimiento en operaciones de procesamiento intensivo.

La arquitectura modular de OpenCV organiza la funcionalidad en varios módulos especializados. El módulo core contiene las estructuras de datos fundamentales y funciones básicas utilizadas por todos los demás módulos. El módulo imgproc implementa algoritmos

de procesamiento de imágenes incluyendo filtrado, transformaciones geométricas, conversión de espacios de color, y detección de características. El módulo imgcodecs maneja la lectura y escritura de imágenes en múltiples formatos. El módulo videoio proporciona capacidades de captura y escritura de video. El módulo highgui ofrece interfaces simples para ventanas gráficas y controles de interfaz de usuario.

Para aplicaciones de reconocimiento facial específicamente, OpenCV incluye el módulo face que implementa varios algoritmos clásicos de reconocimiento incluyendo EigenFaces, FisherFaces y LBPH, junto con funcionalidad para detección de puntos de referencia faciales y estimación de pose. El módulo objdetect contiene clasificadores en cascada preentrenados para detección de rostros, ojos, y otros objetos comunes.

El paradigma de código abierto bajo el cual se desarrolla OpenCV proporciona múltiples ventajas para investigación y desarrollo. La disponibilidad del código fuente permite inspeccionar, modificar y extender los algoritmos según necesidades específicas. La ausencia de costos de licenciamiento elimina barreras económicas para la adopción. La documentación extensa y la comunidad activa facilitan el aprendizaje y la resolución de problemas. La naturaleza multiplataforma permite desarrollar aplicaciones que ejecutan en Windows, Linux, macOS, Android e iOS con la misma base de código.

En el contexto de esta investigación, OpenCV 4.x proporciona la infraestructura fundamental para todas las operaciones de procesamiento de video y reconocimiento facial.

La captura de flujos de video mediante protocolo RTSP se implementa utilizando la clase VideoCapture del módulo videoio. La detección facial emplea el clasificador Haar Cascade haarcascade_frontalface_default.xml del módulo objdetect. El reconocimiento facial se implementa mediante las clases EigenFaceRecognizer, FisherFaceRecognizer y LBPHFaceRecognizer del módulo face. Las operaciones de preprocesamiento como conversión a escala de grises, normalización de histogramas y redimensionamiento utilizan funciones del módulo imgproc.

La decisión de basar el desarrollo en OpenCV en lugar de frameworks de aprendizaje profundo más modernos como TensorFlow o PyTorch responde a consideraciones específicas de eficiencia computacional en hardware de recursos limitados. Los algoritmos clásicos implementados en OpenCV, aunque generalmente menos precisos que modelos de aprendizaje profundo de última generación en benchmarks académicos, ofrecen un balance superior entre precisión y costo computacional para aplicaciones en dispositivos de borde como Raspberry Pi. Mientras que un modelo de red neuronal convolucional profunda podría requerir varios cientos de milisegundos para procesar un solo rostro en Raspberry Pi, los algoritmos clásicos de OpenCV pueden procesar múltiples rostros por segundo, habilitando aplicaciones de tiempo real que serían inviables con arquitecturas más complejas.

1.5. Sistemas de Gestión de Video (VMS) y Protocolo RTSP

Los Sistemas de Gestión de Video (VMS, por sus siglas en inglés Video Management System) constituyen plataformas de software diseñadas para centralizar el control, grabación, visualización y análisis de múltiples flujos de video procedentes de cámaras de vigilancia distribuidas geográficamente. A diferencia de los sistemas de circuito cerrado de televisión (CCTV) tradicionales que operaban mediante conexiones analógicas punto a punto, los VMS modernos aprovechan las redes IP para integrar cámaras de diversos fabricantes en una infraestructura unificada de gestión.

Un VMS típico proporciona múltiples funcionalidades críticas para operaciones de seguridad y vigilancia. La grabación continua o activada por eventos permite almacenar video histórico para análisis forense. La visualización en tiempo real mediante interfaces multipantalla facilita el monitoreo simultáneo de múltiples ubicaciones. La gestión de permisos y control de acceso garantiza que diferentes operadores visualicen únicamente las cámaras autorizadas. Las capacidades de búsqueda y reproducción permiten localizar rápidamente eventos específicos en archivos extensos. La integración con sistemas de alarmas y control de acceso habilita respuestas automáticas a eventos detectados.

Los VMS empresariales comerciales como Milestone XProtect, Genetec Security Center, Avigilon Control Center, y las soluciones propietarias de fabricantes como HikCentral (Hikvision) y SmartPSS (Dahua) ofrecen funcionalidades avanzadas incluyendo análisis de video mediante inteligencia artificial. Estas capacidades incluyen

detección de intrusiones en perímetros definidos, conteo de personas, reconocimiento de matrículas vehiculares, detección de objetos abandonados, análisis de multitudes, y reconocimiento facial. Sin embargo, estas funcionalidades avanzadas frecuentemente requieren licencias adicionales que pueden superar los $400.000 COP por cámara, sin incluir los costos de la licencia base del VMS.

El protocolo RTSP (Real Time Streaming Protocol) representa el estándar de facto para la transmisión de flujos de video en tiempo real sobre redes IP. Definido originalmente en RFC 2326 y posteriormente actualizado en RFC 7826, RTSP funciona como un protocolo de control a nivel de aplicación que establece y controla sesiones de transmisión de medios entre un servidor (típicamente una cámara IP) y un cliente (un VMS o aplicación de visualización).

RTSP opera mediante un mecanismo de solicitud-respuesta similar a HTTP, pero optimizado para controlar flujos de medios en tiempo real. El cliente inicia la comunicación enviando una solicitud DESCRIBE al servidor, especificando la URL del flujo de video deseado. El servidor responde con un descriptor de sesión en formato SDP (Session Description Protocol) que especifica las características del flujo, incluyendo códecs de video y audio, resolución, tasa de bits, y los puertos de red para la transmisión de datos.

Posteriormente, el cliente envía solicitudes SETUP para configurar los canales de transporte, típicamente utilizando RTP (Real-time Transport Protocol) sobre UDP para la

transmisión de paquetes de video con baja latencia. La solicitud PLAY inicia la transmisión del flujo, mientras que PAUSE y TEARDOWN permiten pausar y finalizar la sesión respectivamente. RTSP también soporta solicitudes OPTIONS para consultar las capacidades del servidor y GET_PARAMETER para recuperar parámetros de sesión.

Una ventaja fundamental de RTSP es su naturaleza estandarizada y abierta, permitiendo interoperabilidad entre cámaras y software de diferentes fabricantes. Prácticamente todas las cámaras IP modernas de fabricantes establecidos como Hikvision, Dahua, Axis, Bosch y Hanwha soportan RTSP para streaming de video, aunque frecuentemente también implementan protocolos propietarios que ofrecen funcionalidades adicionales específicas del fabricante.

Las URLs RTSP siguen un formato estandarizado que típicamente incluye credenciales de autenticación, dirección IP de la cámara, puerto (por defecto 554), y una ruta específica del fabricante que identifica el flujo particular. Por ejemplo, las cámaras Hikvision utilizan rutas como rtsp://usuario:contraseña@192.168.1.64:554/Streaming/Channels/101 para el flujo principal de alta resolución y /102 para el subflujo de menor resolución. Dahua emplea rutas como /cam/realmonitor?channel=1&subtype=0, mientras que Axis utiliza /axis-media/media.amp.

En el contexto de sistemas de reconocimiento facial, RTSP proporciona acceso directo a los flujos de video sin necesidad de intermediación del VMS del fabricante. Esto permite implementar análisis de video personalizado mediante software de código abierto, procesando los fotogramas en tiempo real para aplicar algoritmos de detección y reconocimiento facial. Esta capacidad de desacoplar la captura de video del análisis representa un habilitador fundamental para soluciones de bajo costo que evitan las licencias comerciales de VMS.

Sin embargo, RTSP también presenta desafíos técnicos. El uso de UDP para transporte de video puede resultar en pérdida de paquetes en redes congestionadas, manifestándose como artefactos visuales o fotogramas perdidos. La latencia, aunque típicamente baja (100-500ms en redes locales), puede acumularse en sistemas que procesan múltiples flujos simultáneamente. La autenticación básica mediante credenciales en la URL expone las contraseñas a interceptación si no se emplea cifrado adicional mediante RTSPS (RTSP sobre TLS).

El sistema desarrollado en esta investigación implementa captura de múltiples flujos RTSP mediante la clase VideoCapture de OpenCV, que abstrae los detalles del protocolo y proporciona una interfaz uniforme para acceder a fotogramas individuales. La configuración incluye manejo de reconexión automática ante interrupciones de red, ajuste dinámico de buffers para minimizar latencia, y decodificación de video mediante aceleración por hardware cuando está disponible en la plataforma Raspberry Pi.

1.6. Hardware de Bajo Costo para Procesamiento de IA en el Borde

El concepto de edge computing o computación en el borde ha emergido como paradigma arquitectónico que distribuye el procesamiento de datos cerca de las fuentes de generación, en contraste con el modelo tradicional de computación en la nube donde los datos se transmiten a centros de datos centralizados para su análisis. En aplicaciones de video vigilancia y reconocimiento facial, el procesamiento en el borde permite analizar flujos de video localmente, reduciendo el ancho de banda de red requerido, minimizando latencias y mejorando la privacidad al evitar la transmisión de datos biométricos sensibles a servidores remotos.

Raspberry Pi representa una familia de computadoras de placa única desarrolladas por la Fundación Raspberry Pi con el objetivo de promover la educación en ciencias de la computación. Desde el lanzamiento del modelo original en 2012, Raspberry Pi ha evolucionado a través de múltiples generaciones que incrementan progresivamente las capacidades computacionales mientras mantienen un factor de forma compacto y costos accesibles. La combinación de procesador ARM, memoria RAM, conectividad de red, puertos USB, y salidas de video en una placa del tamaño de una tarjeta de crédito, con precios entre $35 USD y $80 USD dependiendo del modelo, ha democratizado el acceso a plataformas computacionales para prototipado y desarrollo de aplicaciones embebidas.

El Raspberry Pi 4 Model B, lanzado en 2019, representa un salto significativo en capacidades respecto a generaciones anteriores. Incorpora un procesador Broadcom BCM2711 con cuatro núcleos ARM Cortex-A72 operando a 1.5 GHz, ofreciendo aproximadamente tres veces el rendimiento de procesamiento del Raspberry Pi 3. Está disponible en configuraciones con 2GB, 4GB u 8GB de memoria RAM LPDDR4, permitiendo ejecutar aplicaciones más exigentes en memoria. Incluye conectividad Gigabit Ethernet real (no limitada por USB como en modelos previos), WiFi 802.11ac de doble banda, Bluetooth 5.0, dos puertos USB 3.0 y dos USB 2.0, dos salidas micro-HDMI que soportan resoluciones hasta 4K, y un conector de cámara CSI para módulos de cámara nativos de Raspberry Pi.

Para aplicaciones de procesamiento de video y reconocimiento facial, el Raspberry Pi 4 demuestra capacidades suficientes para analizar flujos de video de definición estándar en tiempo real. Pruebas realizadas en el contexto de esta investigación confirmaron que un Raspberry Pi 4 de 4GB puede procesar simultáneamente dos flujos de video a 640×480 píxeles y 15 fotogramas por segundo, aplicando detección facial mediante Haar Cascade y reconocimiento mediante LBPH con latencias inferiores a un segundo por identificación. Esta capacidad resulta adecuada para escenarios de control de acceso donde los usuarios se aproximan a la cámara de forma controlada, aunque insuficiente para aplicaciones de vigilancia continua de múltiples individuos en movimiento.

El Raspberry Pi 5, lanzado en octubre de 2023, introduce mejoras arquitectónicas sustanciales que duplican aproximadamente el rendimiento del Raspberry Pi 4. El nuevo procesador Broadcom BCM2712 integra cuatro núcleos ARM Cortex-A76 operando a 2.4 GHz, una arquitectura más moderna que ofrece mayor rendimiento por ciclo de reloj. La GPU VideoCore VII proporciona capacidades gráficas significativamente superiores. La interfaz PCI Express 2.0 habilita conectividad de alta velocidad para periféricos como almacenamiento NVMe. Las mejoras en el subsistema de memoria y los buses de interconexión reducen los cuellos de botella que limitaban el rendimiento en generaciones anteriores.

Las pruebas de capacidad realizadas demostraron que el Raspberry Pi 5 de 8GB puede gestionar exitosamente cuatro flujos de video simultáneos a resolución 640×480 y 20 fotogramas por segundo, manteniendo latencias de reconocimiento facial inferiores a 500 milisegundos. Este incremento en escalabilidad, junto con el costo de aproximadamente $80 USD para el modelo de 8GB, posiciona al Raspberry Pi 5 como una plataforma viable para instalaciones de escala mediana con decenas de puntos de acceso monitoreados.

Una ventaja adicional del ecosistema Raspberry Pi es la disponibilidad del sistema operativo Raspberry Pi OS, una distribución de Linux basada en Debian optimizada específicamente para el hardware Raspberry Pi. Este sistema operativo incluye drivers optimizados, herramientas de configuración simplificadas, y una vasta colección de

software precompilado que incluye Python, OpenCV y otras bibliotecas relevantes para aplicaciones de visión por computadora. La naturaleza de código abierto del sistema operativo y la compatibilidad con el ecosistema completo de software Linux facilitan la personalización y el desarrollo de aplicaciones especializadas.

Es importante reconocer también las limitaciones inherentes al hardware de bajo costo. El rendimiento computacional, aunque suficiente para algoritmos clásicos de reconocimiento facial, resulta insuficiente para modelos de aprendizaje profundo de última generación que requieren cientos de operaciones de punto flotante por píxel. La ausencia de aceleración por GPU para frameworks de deep learning como TensorFlow o PyTorch en arquitecturas ARM limita las opciones algorítmicas. La refrigeración pasiva puede resultar inadecuada bajo cargas sostenidas, requiriendo ventilación forzada o disipadores térmicos más grandes para prevenir el throttling térmico. El almacenamiento en tarjetas microSD, aunque económico, presenta limitaciones en velocidad y durabilidad comparado con SSDs, particularmente problemático para aplicaciones que escriben continuamente logs o graban video.

No obstante, estas limitaciones deben contextualizarse frente al objetivo de democratización tecnológica que motiva esta investigación. El balance entre costo, rendimiento y suficiencia funcional que ofrecen plataformas como Raspberry Pi las posiciona como habilitadores viables para organizaciones que, de otro modo, quedarían excluidas del acceso a tecnologías de reconocimiento facial por barreras económicas.

1.7. Estado del Arte y Trabajos Relacionados

El reconocimiento facial automático ha experimentado una evolución acelerada en las últimas dos décadas, transitando desde sistemas experimentales de laboratorio con tasas de error superiores al 10% en condiciones controladas, hasta soluciones comerciales que alcanzan precisiones superiores al 99% en datasets de referencia como Labeled Faces in the Wild (LFW). Esta sección examina el estado actual del campo, las soluciones comerciales dominantes, y los trabajos académicos relacionados que contextualizan la presente investigación.

En el ámbito comercial, los sistemas VMS con capacidades de reconocimiento facial han alcanzado madurez técnica significativa. Milestone XProtect, posicionado como líder del mercado según múltiples estudios de analistas, ofrece integración con motores de reconocimiento facial de terceros como BriefCam y AnyVision, permitiendo búsquedas retroactivas de individuos específicos en archivos de video históricos y alertas en tiempo real ante detección de personas de interés. Genetec Security Center proporciona funcionalidad similar mediante su módulo AutoVu, originalmente diseñado para reconocimiento de matrículas, pero expandido para incluir reconocimiento facial.

Los fabricantes de cámaras han desarrollado ecosistemas verticalmente integrados que incluyen tanto hardware de captura como software VMS propietario. HikCentral Professional de Hikvision integra reconocimiento facial utilizando cámaras especializadas

de la serie DeepinMind que incorporan procesadores de IA dedicados para análisis en el borde. El sistema puede gestionar bases de datos de hasta 300,000 rostros en configuraciones empresariales, con capacidad de identificación en tiempo real incluso en condiciones de iluminación subóptimas o con accesorios como gafas. SmartPSS de Dahua ofrece capacidades comparables mediante su línea de cámaras WizSense y WizMind.

Estas soluciones comerciales ofrecen ventajas indiscutibles en términos de integración, soporte técnico, garantías de rendimiento y ecosistemas maduros de partners de integración. Sin embargo, los modelos de licenciamiento representan barreras económicas significativas. Una instalación típica de 20 cámaras con capacidades de reconocimiento facial en Milestone XProtect requiere licencias base del VMS más licencias por cámara para el módulo de analítica, sumando inversiones superiores a los $20,000 USD sin incluir hardware, instalación ni capacitación. HikCentral Professional, aunque más económico, aún representa inversiones de varios miles de dólares para instalaciones de escala mediana.

En el ámbito académico, múltiples investigaciones han explorado alternativas de código abierto y hardware de bajo costo para reconocimiento facial. Khan et al. (2022) documentaron la implementación de un sistema de control de asistencia universitaria basado en Raspberry Pi 3 y OpenCV, utilizando el algoritmo LBPH para reconocimiento. Su sistema alcanzó tasas de reconocimiento del 87% en condiciones controladas de iluminación interior, procesando un rostro por segundo. Sin embargo, el trabajo se limitó

a un único flujo de video de cámara USB directamente conectada al Raspberry Pi, sin abordar la escalabilidad a múltiples cámaras o integración con infraestructuras de red IP.

Oliveira y Silva (2021) evaluaron comparativamente el rendimiento de EigenFaces, FisherFaces y LBPH en Raspberry Pi 4, utilizando el dataset Extended Yale B que incluye variaciones extremas de iluminación. Sus resultados indicaron que LBPH superaba consistentemente a los otros algoritmos en condiciones de iluminación variable, alcanzando precisiones del 94% comparado con 78% para EigenFaces y 85% para FisherFaces. Estos hallazgos fundamentan la decisión de priorizar LBPH en el sistema desarrollado en la presente investigación.

Parkhi, Vedaldi y Zisserman (2015) introdujeron VGGFace, un modelo de red neuronal convolucional entrenado con 2.6 millones de imágenes de 2,622 individuos, alcanzando precisiones superiores al 98% en el benchmark LFW. Schroff, Kalenichenko y Philbin (2015) propusieron FaceNet, que utiliza una arquitectura de red siamesa para aprender embeddings de rostros en un espacio euclidiano donde las distancias corresponden directamente a similitud facial, alcanzando 99.63% de precisión en LFW. Estos trabajos representan el estado del arte en términos de precisión, pero requieren GPUs de alto rendimiento para inferencia en tiempo real, haciéndolos inviables para plataformas de bajo costo como Raspberry Pi.

Investigaciones más recientes han explorado la optimización de modelos de aprendizaje profundo para dispositivos de borde. Howard et al. (2017) introdujeron MobileNets, arquitecturas de redes neuronales diseñadas específicamente para aplicaciones móviles y embebidas mediante el uso de convoluciones separables en profundidad que reducen drásticamente el costo computacional. Chang et al. (2023) evaluaron específicamente el rendimiento de MobileNetV2 para reconocimiento facial en Raspberry Pi 5, reportando tiempos de inferencia de 150ms por rostro con precisiones del 95% en condiciones controladas. Estos resultados sugieren que las generaciones futuras de hardware de bajo costo podrían hacer viable el uso de modelos de aprendizaje profundo optimizados.

En el contexto latinoamericano y colombiano específicamente, la literatura académica sobre implementaciones de reconocimiento facial de bajo costo es escasa. La mayoría de las publicaciones se concentran en países con mayor desarrollo de investigación en visión por computadora como Brasil, México y Argentina. Esta carencia representa tanto una limitación en términos de trabajos de referencia directamente aplicables al contexto local, como una oportunidad para contribuir conocimiento relevante para instituciones de la región.

La brecha que aborda la presente investigación se sitúa en la intersección entre capacidad técnica, viabilidad económica y escalabilidad práctica. Mientras que trabajos académicos previos han demostrado la factibilidad técnica de reconocimiento facial en

Raspberry Pi, generalmente se han limitado a pruebas de concepto con cámaras únicas conectadas directamente. Las soluciones comerciales ofrecen escalabilidad y gestión centralizada, pero a costos prohibitivos para instituciones con recursos limitados. Este trabajo desarrolla y evalúa una arquitectura completa que integra múltiples flujos de video mediante protocolo RTSP estándar, gestión centralizada de bases de datos de rostros, y evaluación comparativa de algoritmos de reconocimiento, todo basado en tecnologías de código abierto y hardware de menos de $100 USD por nodo de procesamiento.

Planteamiento del Problema

La seguridad física en entornos de alta circulación representa un desafío operativo y económico creciente para organizaciones de diversos sectores. Centros comerciales, complejos empresariales, instituciones educativas, edificios gubernamentales y hospitales requieren sistemas de control de acceso que garanticen la identificación confiable de personas autorizadas mientras impiden el ingreso de individuos no autorizados. La efectividad de estos sistemas impacta directamente la seguridad de bienes, información sensible y, fundamentalmente, la integridad física de las personas que ocupan estos espacios.

Los métodos tradicionales de control de acceso presentan limitaciones bien documentadas que comprometen tanto su seguridad como su eficiencia operativa. Los

sistemas basados en tarjetas de proximidad o credenciales magnéticas son vulnerables a pérdidas, duplicación no autorizada, y transferencia entre usuarios, eliminando la certeza de que la persona que presenta la credencial es efectivamente su titular legítimo. Los códigos PIN pueden ser compartidos, observados durante su ingreso, u obtenidos mediante ingeniería social. Los registros manuales mediante libros de visitas o planillas dependen de la honestidad del usuario y la vigilancia continua del personal de seguridad, resultando en procesos lentos que generan congestión en puntos de acceso durante horas pico.

Adicionalmente, estos métodos tradicionales imponen costos operativos recurrentes significativos. La emisión, reemplazo por pérdida o deterioro, y administración de credenciales físicas requiere personal dedicado y materiales consumibles. Los sistemas de tarjetas exigen infraestructura de lectores, cableado, y controladores de acceso en cada punto de entrada. El personal de seguridad debe dedicar tiempo a verificar identidades, registrar accesos, y gestionar situaciones de credenciales olvidadas o visitantes no registrados. Para una organización con cientos de empleados y flujo constante de visitantes, estos costos administrativos pueden alcanzar varios millones de pesos anuales.

La identificación biométrica, específicamente el reconocimiento facial, ha emergido como alternativa tecnológica que aborda muchas de estas limitaciones. Al vincular la identidad directamente con características físicas únicas e intransferibles del rostro humano, elimina la posibilidad de transferencia de credenciales. Su naturaleza no invasiva y sin contacto permite identificaciones rápidas sin requerir que los usuarios

detengan su marcha, presenten tarjetas, o interactúen con dispositivos, agilizando el flujo en puntos de acceso. La capacidad de operar a distancia posibilita identificaciones discretas sin cooperación activa del usuario, útil en escenarios de vigilancia preventiva.

Sin embargo, la adopción de tecnologías de reconocimiento facial enfrenta una barrera económica crítica en el contexto colombiano. Los Sistemas de Gestión de Video Empresarial (VMS) comerciales que integran capacidades de identificación facial imponen estructuras de licenciamiento que resultan prohibitivas para instituciones con recursos limitados. Las soluciones de fabricantes establecidos como Milestone XProtect, Genetec Security Center, o Avigilon Control Center requieren licencias base del VMS más módulos adicionales de analítica por cámara. El costo de licenciamiento para funcionalidades de reconocimiento facial puede superar los $400.000 COP por cámara, sin incluir costos de hardware, instalación profesional, capacitación de operadores, ni mantenimiento anual.

Para contextualizar la magnitud de esta barrera, considérese una institución educativa de tamaño medio que requiere controlar accesos en 15 puntos de entrada distribuidos en su campus. Implementar reconocimiento facial mediante una solución VMS comercial implicaría inversiones iniciales superiores a $15.000.000 COP únicamente en licencias de software, cantidad que excede los presupuestos de seguridad anuales de muchas instituciones públicas y organizaciones sin fines de lucro. Pequeñas y medianas empresas que podrían beneficiarse significativamente de sistemas automatizados de control de acceso quedan efectivamente excluidas de estas tecnologías.

Existen alternativas de menor costo en el mercado, específicamente terminales de identificación facial independientes fabricados por empresas como ZKTeco, Anviz o Suprema. Estos dispositivos, con precios entre $800.000 y $1.500.000 COP por unidad, ofrecen funcionalidades básicas de reconocimiento facial y control de acceso. Sin embargo, operan como sistemas cerrados que carecen de las capacidades fundamentales que requieren instalaciones de mediana y gran escala. No proporcionan gestión centralizada que permita administrar múltiples puntos de acceso desde una consola unificada. No se integran con infraestructuras de cámaras IP existentes, requiriendo reemplazo completo de equipamiento de video vigilancia. No permiten búsquedas retroactivas en video histórico para localizar apariciones previas de individuos específicos. Su escalabilidad está limitada por diseño, típicamente soportando bases de datos de hasta 3.000 rostros y configuraciones con menos de 10 terminales interconectadas.

Esta situación configura una brecha tecnológica donde únicamente las grandes corporaciones con presupuestos sustanciales pueden implementar sistemas de reconocimiento facial con gestión centralizada y escalabilidad empresarial, mientras que instituciones educativas, pequeñas y medianas empresas, organizaciones sin fines de lucro, y entidades gubernamentales con restricciones presupuestarias quedan relegadas a métodos tradicionales vulnerables o soluciones de reconocimiento facial aisladas sin capacidades de integración.

La problemática se agrava considerando que muchas de estas instituciones excluidas operan precisamente en contextos donde la seguridad robusta es crítica. Colegios y universidades albergan poblaciones vulnerables de menores de edad. Hospitales manejan información médica sensible y medicamentos controlados. Edificios municipales procesan documentación ciudadana confidencial. Pequeñas empresas de sectores como tecnología o servicios financieros custodian información competitiva o datos personales de clientes. La imposibilidad de acceder a tecnologías modernas de control de acceso perpetúa vulnerabilidades de seguridad con potenciales consecuencias severas.

Más allá de las consideraciones de seguridad, existe un componente de equidad tecnológica. El acceso diferenciado a tecnologías de seguridad basado en capacidad económica profundiza brechas existentes entre organizaciones con recursos abundantes y aquellas con limitaciones presupuestarias. Las instituciones excluidas no solo enfrentan mayores riesgos de seguridad, sino también desventajas competitivas en mercados donde la capacidad de garantizar seguridad de información y activos influye en decisiones de clientes y partners.

La disponibilidad de tecnologías de código abierto y hardware de bajo costo sugiere que esta brecha no es inevitable ni tecnológicamente justificada. Bibliotecas como OpenCV proporcionan implementaciones maduras de algoritmos de reconocimiento facial sin costos de licenciamiento. El protocolo RTSP permite integración con cámaras IP de múltiples fabricantes sin depender de software propietario. Plataformas de hardware como

Raspberry Pi ofrecen capacidades computacionales suficientes para procesamiento de video en tiempo real a costos inferiores a $100 USD por nodo. La existencia de estos componentes abiertos y accesibles plantea la pregunta de si es técnica y económicamente viable integrarlos en una solución funcional que proporcione las capacidades fundamentales de gestión centralizada y escalabilidad que requieren las instalaciones modernas.

Esta investigación aborda precisamente esta pregunta, planteándola formalmente de la siguiente manera:

¿Cómo diseñar y evaluar una arquitectura abierta de identificación facial que se integre con Sistemas de Gestión de Video mediante protocolo RTSP, utilizando exclusivamente tecnologías de código abierto y hardware de bajo costo, garantizando viabilidad económica, escalabilidad técnica y gestión centralizada para instituciones con recursos limitados?

Justificación

La presente investigación se justifica desde múltiples perspectivas que abarcan dimensiones técnicas, económicas, sociales y académicas, configurando un aporte significativo tanto para el conocimiento científico como para la práctica profesional en contextos reales.

Desde la perspectiva técnica, este trabajo contribuye al cuerpo de conocimiento sobre implementación de sistemas de reconocimiento facial en plataformas de recursos computacionales limitados. Mientras que la literatura académica abunda en estudios sobre

algoritmos de reconocimiento facial de última generación ejecutando en servidores con GPUs de alto rendimiento, existe escasez de investigaciones que documenten sistemáticamente el desempeño, limitaciones y optimizaciones necesarias para implementar estos sistemas en hardware de propósito general de bajo costo. La evaluación comparativa de algoritmos clásicos de reconocimiento facial (EigenFaces, FisherFaces, LBPH) específicamente en plataformas Raspberry Pi proporciona información práctica valiosa sobre los balances entre precisión, velocidad de procesamiento y consumo de recursos que enfrentan los desarrolladores de aplicaciones embebidas.

La arquitectura de software desarrollada, que integra captura de múltiples flujos de video mediante protocolo RTSP, procesamiento distribuido en nodos de bajo costo, y gestión centralizada de bases de datos de rostros, representa un aporte metodológico replicable. La documentación detallada del proceso de diseño, las decisiones arquitectónicas, y las soluciones a desafíos específicos como manejo de reconexiones de red, sincronización de flujos múltiples, y optimización de latencias, constituye conocimiento transferible que otros investigadores y desarrolladores pueden adaptar a sus contextos particulares.

Desde la perspectiva económica, el impacto potencial de esta investigación trasciende el ámbito puramente académico para tener relevancia práctica directa. La demostración de que es viable desarrollar sistemas de identificación facial funcionales con inversiones inferiores al 15% del costo de soluciones comerciales equivalentes abre

posibilidades concretas para instituciones que actualmente están excluidas de estas tecnologías por barreras económicas. Una institución educativa, una PYME del sector servicios, o una entidad gubernamental municipal que no puede justificar una inversión de $15.000.000 COP en licencias de VMS comercial, podría considerar viable una inversión de $2.000.000 COP en hardware de bajo costo y desarrollo personalizado basado en las metodologías documentadas en este trabajo.

La viabilidad económica no se limita a costos iniciales de adquisición. Los sistemas basados en código abierto eliminan costos recurrentes de licenciamiento anual, renovaciones de soporte técnico, y actualizaciones de versiones que caracterizan a las soluciones comerciales propietarias. Para una organización que planifica operación del sistema durante 5-10 años, la ausencia de estos costos recurrentes representa ahorros acumulados que pueden superar varias veces la inversión inicial, mejorando sustancialmente el retorno de inversión y la sostenibilidad financiera a largo plazo.

Desde la perspectiva social, esta investigación se alinea con principios de democratización del acceso a tecnologías avanzadas. La concentración del acceso a sistemas de seguridad modernos en organizaciones con recursos económicos abundantes perpetúa inequidades donde instituciones que sirven a poblaciones vulnerables o comunidades con menores recursos operan con infraestructuras de seguridad obsoletas. Instituciones educativas públicas en municipios con presupuestos limitados, centros de salud rurales, bibliotecas públicas, o centros comunitarios podrían implementar control de

acceso automatizado mejorando la seguridad de sus usuarios sin comprometer recursos que debieran destinarse a su misión principal.

La naturaleza de código abierto del desarrollo promueve además una cultura de transparencia y auditabilidad que contrasta con los sistemas propietarios tipo "caja negra". Los algoritmos, las bases de datos, y los procesos de toma de decisiones son inspeccionables, permitiendo que las organizaciones que implementan estos sistemas comprendan exactamente cómo funcionan, qué datos almacenan, y cómo se utilizan. Esta transparencia es particularmente relevante en el contexto de tecnologías biométricas que procesan datos personales sensibles, donde las preocupaciones sobre privacidad y uso ético de información son legítimas y merecen mecanismos de accountability.

Desde la perspectiva de formación profesional, el desarrollo de este proyecto proporciona experiencia práctica integral en múltiples dominios técnicos relevantes para la ingeniería de sistemas contemporánea. La integración de visión por computadora, procesamiento en tiempo real, arquitecturas distribuidas, protocolos de red, programación de sistemas embebidos, y optimización de rendimiento en entornos de recursos limitados constituye un ejercicio formativo que desarrolla competencias directamente transferibles al ejercicio profesional. Las habilidades adquiridas en la evaluación comparativa de algoritmos, el diseño de arquitecturas escalables, y la documentación técnica rigurosa son aplicables a dominios diversos más allá del reconocimiento facial específicamente.

En el contexto regional del Eje Cafetero y específicamente del departamento del Quindío, esta investigación responde a necesidades locales identificables. La región alberga múltiples instituciones educativas, centros comerciales de mediana escala, entidades gubernamentales municipales, y un sector empresarial predominantemente compuesto por pequeñas y medianas empresas que podrían beneficiarse de sistemas automatizados de control de acceso pero que enfrentan las barreras económicas descritas.

Finalmente, desde una perspectiva de sostenibilidad a largo plazo, la arquitectura basada en estándares abiertos y hardware de propósito general reduce riesgos de obsolescencia planificada y dependencia de proveedores específicos. Los sistemas propietarios frecuentemente discontinúan soporte para versiones antiguas, forzando actualizaciones costosas o reemplazos completos de equipamiento funcional. Una arquitectura basada en protocolos estándares como RTSP y bibliotecas de código abierto como OpenCV mantiene compatibilidad con equipamiento diverso y tiene probabilidades mayores de permanecer funcional y actualizable a lo largo de horizontes temporales extensos.

El impacto esperado de esta investigación incluye la generación de conocimiento técnico documentado y replicable sobre implementación de sistemas de reconocimiento facial en plataformas de bajo costo, la demostración empírica de viabilidad económica mediante comparación cuantitativa con soluciones comerciales, la provisión de una arquitectura de referencia que otras instituciones puedan adaptar a sus contextos

específicos, y la contribución a la formación de recurso humano capacitado en tecnologías de visión por computadora y sistemas embebidos con orientación hacia aplicaciones prácticas de impacto social.

Objetivos

Objetivo General

Diseñar y evaluar una arquitectura abierta de identificación facial orientada a su implementación en Sistemas de Gestión de Video Empresarial (VMS), utilizando exclusivamente tecnologías de código abierto y hardware de bajo costo, garantizando viabilidad económica, accesibilidad tecnológica y escalabilidad técnica para instituciones con recursos limitados.

Objetivos específicos

- 1. Diseñar la arquitectura de software de código abierto para la gestión de video. Especificar la integración de algoritmos de identificación facial (EigenFaces, FisherFaces y LBPH) mediante la biblioteca OpenCV, la captura de flujos de video utilizando el protocolo RTSP para compatibilidad con cámaras IP de múltiples fabricantes, y la estructura de bases de datos para gestión centralizada de rostros autorizados.

- 2. Implementar y configurar la infraestructura de hardware de bajo costo. Desplegar el sistema en plataformas Raspberry Pi 4 y Raspberry Pi 5, evaluando la capacidad de procesamiento simultáneo de múltiples flujos de video,

y validando la compatibilidad con cámaras de diferentes tecnologías (USB, IP con protocolo RTSP, y cámaras análogas mediante conversores IP) de fabricantes como Hikvision, Dahua y Hilook.

- 3. Evaluar el rendimiento técnico y la viabilidad económica del sistema. Medir la precisión de identificación facial utilizando datasets de referencia y pruebas con usuarios reales, analizar los tiempos de respuesta y la escalabilidad del sistema en función del número de flujos de video procesados simultáneamente, y comparar cuantitativamente los costos de implementación frente a soluciones VMS comerciales equivalentes, documentando la reducción porcentual de inversión requerida.

Metodología

Tipo y diseño de investigación

La presente investigación se enmarca en el paradigma de investigación aplicada con enfoque cuantitativo y diseño experimental en entorno controlado. La investigación aplicada se caracteriza por su orientación hacia la solución de problemas prácticos específicos mediante la aplicación de conocimientos científicos existentes, en contraste con la investigación básica que busca primordialmente expandir el conocimiento teórico sin consideraciones inmediatas de aplicabilidad. En este caso, el problema práctico abordado es la inaccesibilidad económica de sistemas de reconocimiento facial con gestión centralizada para instituciones con recursos limitados.

El enfoque cuantitativo se manifiesta en la recolección y análisis de datos numéricos objetivos que permiten evaluar el desempeño del sistema desarrollado. Métricas como tasas de reconocimiento correctas, tasas de falsos positivos y falsos negativos, tiempos de procesamiento medidos en milisegundos, número de flujos de video procesados simultáneamente, y costos monetarios expresados en pesos colombianos constituyen variables cuantificables que facilitan comparaciones objetivas y conclusiones basadas en evidencia empírica.

El diseño experimental se implementa mediante la construcción de un prototipo funcional del sistema propuesto, la ejecución de pruebas controladas bajo condiciones

definidas, y la medición sistemática de variables de desempeño. El control de variables se logra manteniendo constantes factores como las condiciones de iluminación en el entorno de pruebas, la resolución y tasa de fotogramas de las cámaras utilizadas, y las características demográficas del conjunto de usuarios participantes en las pruebas de reconocimiento.

Fases del desarrollo

El desarrollo del sistema se estructuró en cuatro fases secuenciales que permitieron avanzar desde la conceptualización teórica hasta la validación empírica de resultados.

- Fase 1: Diseño de la arquitectura del sistema. Esta fase inicial comprendió la definición de los componentes arquitectónicos del sistema, sus interrelaciones, y las tecnologías específicas a emplear. Se especificó la arquitectura de software identificando los módulos de captura de video mediante RTSP, el módulo de detección facial utilizando clasificadores Haar Cascade, el módulo de reconocimiento facial implementando los tres algoritmos seleccionados (EigenFaces, FisherFaces y LBPH), y el módulo de gestión de base de datos de rostros autorizados. Se definió la arquitectura de hardware especificando las plataformas de procesamiento (Raspberry Pi 4 y 5), los tipos de cámaras compatibles, y la topología de red para interconexión de componentes.

Se elaboraron diagramas de arquitectura que representan la estructura modular del sistema, diagramas de flujo que documentan los procesos de entrenamiento y

reconocimiento, y especificaciones técnicas de las interfaces entre componentes. Esta documentación de diseño sirvió como guía para la implementación posterior y proporciona información replicable para otras instituciones que deseen adaptar la solución a sus contextos específicos.

- Fase 2: Implementación del sistema. La segunda fase materializó el diseño mediante la programación de los componentes de software y la configuración de la infraestructura de hardware. Se instaló el sistema operativo Raspbian (distribución de Linux basada en Debian optimizada para Raspberry Pi) en las plataformas de procesamiento. Se configuró el entorno de desarrollo con Python 3.8 como lenguaje de programación principal, aprovechando su sintaxis clara y el amplio ecosistema de bibliotecas para visión por computadora.

Se instaló OpenCV 4.5 mediante gestores de paquetes y compilación desde código fuente para habilitar optimizaciones específicas de la arquitectura ARM de Raspberry Pi. Se desarrollaron los scripts de Python que implementan la captura de video desde URLs RTSP, la detección facial mediante el clasificador preentrenado haarcascade_frontalface_default.xml, y el reconocimiento facial utilizando las clases EigenFaceRecognizer, FisherFaceRecognizer y LBPHFaceRecognizer del módulo face de OpenCV.

Se implementó el módulo de captura de rostros para entrenamiento (capturandoRostros.py) que permite registrar nuevos usuarios capturando 150 imágenes de su rostro desde diferentes ángulos y expresiones. El módulo de entrenamiento (entrenamientoRF.py) procesa las imágenes capturadas, extrae características faciales, y entrena los modelos de reconocimiento generando archivos de modelo persistentes (modeloEigenFace.xml, modeloFisherFace.xml, modeloLBPHFace.xml). El módulo de reconocimiento en tiempo real (ReconocimientoFacial.py) carga los modelos entrenados, captura video en tiempo real, detecta rostros, y realiza identificaciones comparando contra la base de datos de rostros conocidos.

Se configuró la integración con cámaras IP de diferentes fabricantes (Hikvision, Dahua, Hilook) mediante sus URLs RTSP específicas. Se validó la compatibilidad con cámaras USB conectadas directamente a los puertos del Raspberry Pi. Se implementó manejo de errores para reconexión automática ante interrupciones de flujos de video, y optimizaciones de buffers para minimizar latencia.

- Fase 3: Pruebas y validación. La tercera fase evaluó sistemáticamente el desempeño del sistema implementado mediante pruebas en condiciones controladas. Se estableció un entorno de laboratorio con iluminación artificial consistente, fondo neutral, y distancias definidas entre las cámaras y los sujetos. Se reclutaron usuarios voluntarios (entre 6 y 10 personas) que participaron

en las pruebas proporcionando consentimiento informado conforme a consideraciones éticas de manejo de datos biométricos.

Para cada usuario, se capturaron 450 imágenes de entrenamiento mediante el módulo de captura, distribuyendo las tomas entre diferentes expresiones faciales (neutra, sonriente), orientaciones de cabeza (frontal, leves rotaciones), y uso de accesorios comunes (gafas, sin gafas). Estas imágenes se utilizaron para entrenar los tres algoritmos de reconocimiento, generando modelos específicos para cada enfoque.

Se ejecutaron pruebas de reconocimiento donde cada usuario se presentaba ante las cámaras en condiciones similares a las de entrenamiento y también con variaciones deliberadas (diferente iluminación, distintas expresiones, con y sin gafas) para evaluar robustez. Se registraron métricas de precisión calculando el porcentaje de identificaciones correctas, falsos positivos (sistema identifica incorrectamente a una persona como otra), y falsos negativos (sistema no reconoce a una persona registrada).

Se realizaron pruebas de escalabilidad evaluando el número máximo de flujos de video que cada plataforma de hardware podía procesar simultáneamente sin degradación inaceptable del rendimiento. Para Raspberry Pi 4, se configuraron gradualmente 1, 2 y 3 flujos de video monitoreando la utilización de CPU, memoria RAM, y latencias de reconocimiento. Para Raspberry Pi 5, se repitió el proceso escalando hasta 4 flujos simultáneos.

Se validó la compatibilidad con el dataset público Labeled Faces in the Wild (LFW), utilizando un subconjunto de imágenes para evaluar el rendimiento del sistema en condiciones estandarizadas que permiten comparación con trabajos previos reportados en la literatura. Los resultados de precisión obtenidos con LFW proporcionan una métrica de referencia independiente del conjunto de usuarios locales.

- Fase 4: Análisis comparativo de costos. La fase final cuantificó la viabilidad económica mediante un análisis comparativo detallado entre los costos de implementación del sistema desarrollado y los costos de soluciones VMS comerciales equivalentes. Se recopiló información de precios de licencias de software VMS como Milestone XProtect, Genetec Security Center, y HikCentral Professional mediante consultas a distribuidores autorizados y cotizaciones oficiales.

Se documentaron los costos de hardware del sistema desarrollado, incluyendo Raspberry Pi 4 (4GB) a aproximadamente $250.000 COP, Raspberry Pi 5 (8GB) a aproximadamente $320.000 COP, tarjetas microSD de 64GB a $40.000 COP, fuentes de alimentación oficiales a $50.000 COP, y gabinetes protectores a $30.000 COP. Se calculó el costo total por nodo de procesamiento sumando estos componentes.

Se establecieron escenarios de implementación representativos (10 cámaras, 20 cámaras, 50 cámaras) y se calculó el costo total de propiedad para cada escenario bajo ambos enfoques: VMS comercial con licenciamiento por cámara, versus sistema de código abierto con nodos Raspberry Pi procesando múltiples cámaras cada uno. Se calcularon porcentajes de reducción de costos y retornos de inversión considerando horizontes temporales de 3 y 5 años.

Población y muestra

Aunque el sistema está orientado a su eventual implementación en grandes superficies con poblaciones de usuarios en cientos o miles de individuos, las limitaciones de un proyecto de investigación académica requirieron evaluar el prototipo en un entorno controlado de laboratorio con una muestra reducida de participantes.

La muestra estuvo conformada por entre 6 y 10 usuarios voluntarios reclutados entre estudiantes y personal de la Universidad Remington sede armenia. Los criterios de inclusión especificaron participantes mayores de edad, sin impedimentos para proporcionar consentimiento informado, y dispuestos a que sus datos biométricos faciales fueran capturados y procesados exclusivamente con fines de investigación académica. Se garantizó a los participantes que las imágenes y datos biométricos no serían utilizados para propósitos distintos a esta investigación, no serían compartidos con terceros, y serían eliminados al concluir el proyecto.

El tamaño limitado de la muestra se justifica considerando que el objetivo primario de esta investigación es demostrar viabilidad técnica y económica del enfoque propuesto mediante una prueba de concepto (Easterbrook, S., Singer, J., Storey, M. A., & Damian, D. (2008), más que desarrollar un sistema de precisión óptima para producción. Las conclusiones sobre rendimiento y escalabilidad técnica derivadas de esta muestra son generalizables dado que las características computacionales del hardware y los algoritmos son independientes del tamaño de la base de datos de usuarios, aunque la precisión absoluta de reconocimiento sí tendería a degradarse en bases de datos significativamente más grandes debido al incremento de oportunidades para confusiones entre individuos similares.

Hardware y software utilizados

- Hardware de procesamiento: Se utilizaron dos plataformas de la familia Raspberry Pi para evaluar capacidades de procesamiento en diferentes rangos de costo y rendimiento. El Raspberry Pi 4 Model B en su configuración de 4GB de RAM proporcionó una plataforma de costo reducido (aproximadamente $250.000 COP) suficiente para aplicaciones de escala limitada. El Raspberry Pi 5 en su configuración de 8GB de RAM ofreció capacidades superiores (aproximadamente $320.000 COP) permitiendo escalabilidad mejorada.

- Dispositivos de captura de video: Se emplearon cámaras de múltiples tecnologías y fabricantes para validar la versatilidad del sistema. Cámaras IP Hikvision modelo DS-2CD2T23G0-I5 con resolución 1920×1080 y soporte

RTSP. Cámaras IP Dahua modelo IPC-HFW1230S con resolución 1920×1080 y streaming RTSP. Cámaras Hilook (marca económica de Hikvision) modelo IPC-B120H con resolución 1280×720. Cámaras USB genéricas con resolución 640×480 conectadas directamente a puertos USB del Raspberry Pi para validación de compatibilidad con dispositivos de bajo costo.

- Sistema operativo: Raspberry Pi OS (anteriormente Raspbian), distribución de Linux basada en Debian 11 (Bullseye) optimizada para hardware Raspberry Pi. Versión de 64 bits para aprovechar completamente las capacidades del procesador ARM de 64 bits en Raspberry Pi 4 y 5.

- Lenguaje de programación: Python 3.8, seleccionado por su sintaxis clara, amplio soporte de bibliotecas de visión por computadora, y facilidad de prototipado rápido. Python permite desarrollo ágil de aplicaciones complejas con menor cantidad de líneas de código comparado con lenguajes compilados como C++, facilitando mantenibilidad y modificaciones futuras.

- Bibliotecas de visión por computadora: OpenCV 4.5 (Open Source Computer Vision Library), biblioteca de código abierto que proporciona más de 2,500 algoritmos optimizados para procesamiento de imágenes y video. Los módulos específicos utilizados incluyen cv2.VideoCapture para captura de flujos RTSP y USB, cv2.CascadeClassifier para detección facial mediante Haar Cascades,

cv2.face.EigenFaceRecognizer, cv2.face.FisherFaceRecognizer y cv2.face.LBPHFaceRecognizer para reconocimiento facial.

- Bibliotecas auxiliares: NumPy para operaciones matriciales eficientes sobre datos de imágenes, imutils para funciones de conveniencia en procesamiento de video como redimensionamiento y rotación, OS para manejo de sistema de archivos y rutas de directorios.

- Almacenamiento: Tarjetas microSD clase 10 U3 de 64GB para almacenamiento del sistema operativo, aplicaciones, y base de datos de rostros. Aunque las tarjetas microSD presentan limitaciones de velocidad comparadas con SSDs, su bajo costo (aproximadamente $40.000 COP) y suficiencia para las tasas de escritura requeridas justificaron su selección.

- Infraestructura de red: Switch Gigabit Ethernet para interconexión de cámaras IP y nodos de procesamiento Raspberry Pi, router WiFi para acceso remoto a las interfaces de administración, cableado de red categoría 5e para conexiones cableadas de baja latencia.

Técnicas de recolección de datos

Los datos se recolectaron directamente desde el funcionamiento operacional del sistema mediante instrumentación programática. El módulo de reconocimiento registró para cada intento de identificación un timestamp indicando fecha y hora, el identificador

de la cámara o flujo de video que capturó el rostro, el resultado de la identificación (nombre del usuario reconocido o "desconocido" si no se encontró coincidencia), el nivel de confianza expresado como valor numérico donde valores menores indican mayor confianza en la identificación, y el tiempo de procesamiento en milisegundos desde la captura del fotograma hasta la obtención del resultado de reconocimiento.

Estos datos se almacenaron en archivos de log estructurados que posteriormente se procesaron mediante scripts de Python para calcular métricas agregadas como tasas de precisión globales, distribuciones de tiempos de procesamiento, y análisis de casos de falsos positivos y falsos negativos.

Las métricas de utilización de recursos del sistema (porcentaje de uso de CPU, memoria RAM disponible, temperatura del procesador) se monitorearon mediante herramientas del sistema operativo Linux como top, htop, y vcgencmd, registrando mediciones periódicas durante las sesiones de prueba para identificar potenciales cuellos de botella o condiciones de throttling térmico que pudieran afectar el rendimiento.

Resultados y Discusión

Esta sección presenta los resultados obtenidos durante las fases de implementación, pruebas y validación del sistema de identificación facial de bajo costo. Los hallazgos se organizan en cuatro subsecciones que abordan respectivamente la implementación exitosa de la arquitectura, el desempeño de los algoritmos de reconocimiento facial, la escalabilidad y capacidades de procesamiento del hardware, y el análisis comparativo de viabilidad económica.

Implementación de la arquitectura del sistema

La arquitectura propuesta se implementó exitosamente integrando los componentes de software de código abierto con las plataformas de hardware de bajo costo especificadas en la metodología. El sistema resultante demostró capacidad funcional para capturar flujos de video mediante protocolo RTSP, detectar rostros en tiempo real, y realizar identificaciones comparando contra una base de datos de rostros previamente registrados.

La Figura 1 documenta el proceso de instalación del sistema operativo base sobre el cual se construyó la plataforma de desarrollo. Se utilizó Debian 11 (Bullseye) como distribución de Linux, posteriormente replicando la configuración en Raspberry Pi OS para los nodos de procesamiento en hardware de bajo costo. Este proceso de instalación requirió aproximadamente 45 minutos por dispositivo, incluyendo la configuración inicial del sistema, actualización de paquetes, e instalación de dependencias.

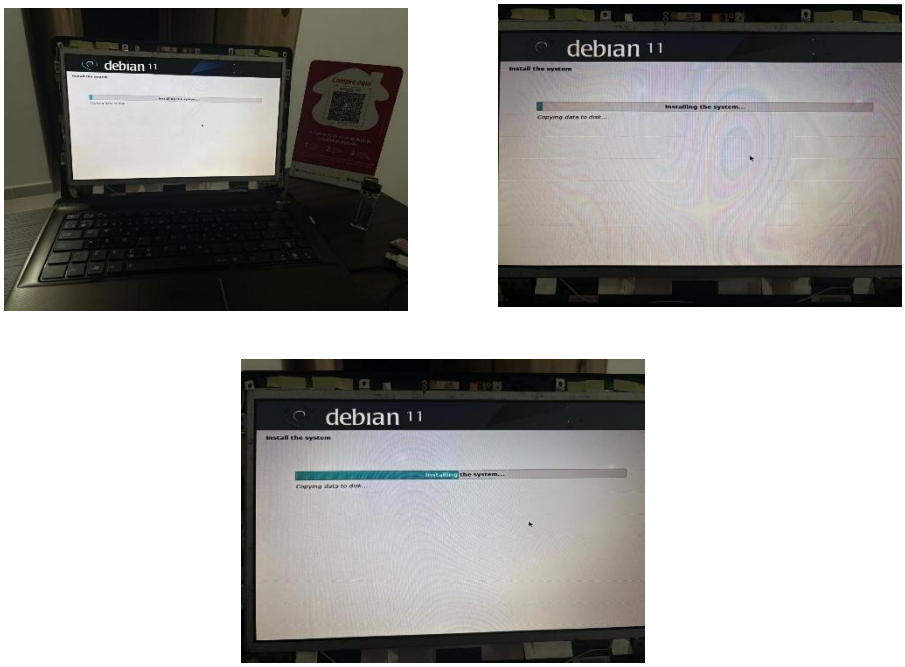

Figura 1. Proceso de instalación del sistema operativo Debian 11 sobre plataforma de desarrollo.

Una vez establecido el sistema operativo base, se procedió con la configuración del entorno Raspberry Pi. La Figura 2 muestra la interfaz del escritorio Raspberry Pi OS completamente configurado y operativo. La familiaridad de la interfaz gráfica basada en el entorno de escritorio LXDE facilita la administración del sistema incluso para usuarios sin experiencia extensa en sistemas Linux.

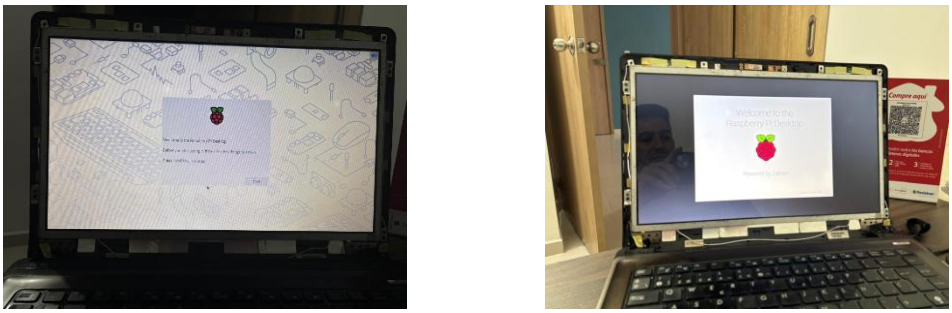

Figura 2. Raspberry Pi Desktop operativo con sistema Raspbian basado en Debian, mostrando el entorno de escritorio LXDE.

La configuración de red representa un aspecto crítico para sistemas que deben integrarse con infraestructuras de cámaras IP existentes. La Figura 3 documenta la configuración de servicios de red mediante Samba, protocolo que permite compartir archivos y recursos entre sistemas Linux y Windows. Esta funcionalidad facilita la administración remota del sistema, permitiendo transferir scripts de Python, actualizar modelos de reconocimiento entrenados, y acceder a logs de operación sin requerir conexión física directa al dispositivo Raspberry Pi.

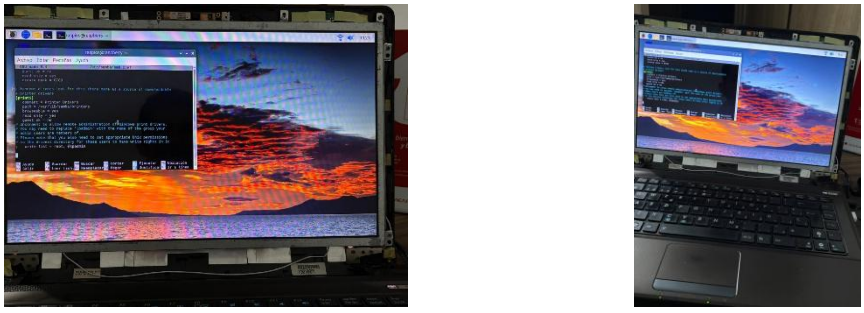

Figura 3. Configuración de servicios de red mediante protocolo Samba para administración remota del sistema.

La Figura 4 ilustra el proceso de creación de cuentas de usuario en el sistema Raspberry Pi. La configuración apropiada de permisos y credenciales de acceso constituye una consideración de seguridad fundamental, especialmente en sistemas que procesarán datos biométricos sensibles. Se implementaron políticas que restringen el acceso a las bases de datos de rostros y los logs de identificación únicamente a usuarios autorizados con credenciales válidas.

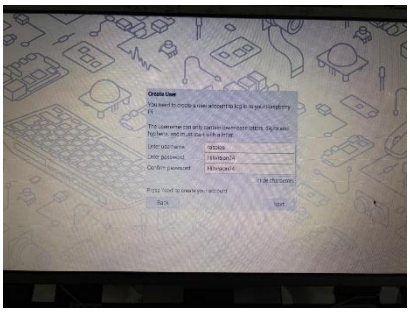

Figura 4. Configuración de credenciales de usuario y políticas de seguridad en Raspberry Pi.

La arquitectura de software implementada sigue un diseño modular que separa las responsabilidades en componentes especializados. La Figura 5 presenta un diagrama conceptual de la arquitectura completa, ilustrando el flujo de datos desde las cámaras IP a través del protocolo RTSP, la detección facial mediante clasificadores Haar Cascade, el reconocimiento mediante los algoritmos entrenados, y la respuesta del sistema indicando la identidad identificada o señalando un rostro desconocido.

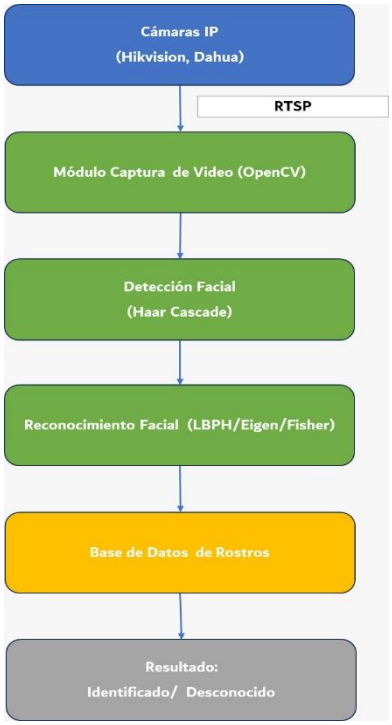

Figura 5. *Arquitectura modular del sistema de identificación facial implementado.*

El módulo de captura de rostros para entrenamiento permite registrar nuevos usuarios en el sistema de forma sencilla. El proceso captura automáticamente 150 fotogramas del rostro del usuario desde diferentes ángulos y expresiones mientras la persona se mueve naturalmente frente a la cámara. Estas variaciones en los datos de entrenamiento mejoran la robustez del reconocimiento posterior ante cambios en la pose, expresión facial, e iluminación.

El módulo de entrenamiento procesa las imágenes capturadas aplicando el algoritmo de reconocimiento seleccionado (EigenFaces, FisherFaces o LBPH) y genera un modelo persistente almacenado en formato XML. Este modelo codifica las características distintivas de los rostros conocidos de manera que permita comparaciones eficientes durante el reconocimiento en tiempo real. El tiempo de entrenamiento varió entre 2 minutos para LBPH y 8 minutos para FisherFaces en Raspberry Pi 4, dependiendo del número de usuarios registrados y la complejidad computacional del algoritmo.

Desempeño de algoritmos de reconocimiento facial

Se evaluaron comparativamente tres algoritmos clásicos de reconocimiento facial: EigenFaces basado en Análisis de Componentes Principales, FisherFaces basado en Análisis Discriminante Lineal, y LBPH basado en patrones binarios locales. Cada algoritmo presenta características distintas en términos de precisión, velocidad de procesamiento, y robustez ante variaciones de iluminación.

Las pruebas se realizaron en dos conjuntos de datos distintos para proporcionar evaluación tanto en condiciones controladas estandarizadas como en condiciones reales con usuarios voluntarios locales.

Evaluación con dataset Labeled Faces in the Wild

Se utilizó un subconjunto del dataset público Labeled Faces in the Wild (LFW) para evaluar el sistema en condiciones estandarizadas que permiten comparación con

trabajos previos en la literatura. LFW contiene más de 13,000 imágenes de rostros de personalidades públicas recolectadas de internet, presentando variaciones naturales de iluminación, pose, expresión y fondo que representan condiciones desafiantes para sistemas de reconocimiento.

Dado que procesar el dataset completo excede las capacidades de memoria y tiempo de procesamiento disponibles en Raspberry Pi, se seleccionó un subconjunto de 500 imágenes correspondientes a 50 individuos (10 imágenes por persona). Se utilizaron 7 imágenes por persona para entrenamiento y 3 para pruebas de reconocimiento, siguiendo protocolos estándares de validación.

Los resultados obtenidos se presentan en la Tabla 1, donde se observa que LBPH alcanzó la mayor precisión con 92% de identificaciones correctas, seguido de FisherFaces con 85% y EigenFaces con 78%. Estos resultados son consistentes con los reportados por Oliveira y Silva (2021) quienes encontraron superioridad de LBPH en condiciones de iluminación variable.

Tabla 1. Comparación de precisión de algoritmos de reconocimiento facial sobre dataset LFW.

Algoritmo	Precisión (%)	Falsos Positivos (%)	Falsos Negativos (%)	Tiempo promedio de reconocimiento (ms)
EigenFaces	78	12	10	145
FisherFaces	85	8	7	180
LBPH	92	5	3	95

El desempeño superior de LBPH se atribuye a su naturaleza local que analiza texturas en regiones pequeñas del rostro, en contraste con los enfoques holísticos de EigenFaces y FisherFaces que analizan el rostro completo. Esta característica hace a

LBPH más robusto ante variaciones locales de iluminación que afectan solamente porciones del rostro, fenómeno común en imágenes capturadas en condiciones no controladas como las de LFW.

Adicionalmente, LBPH demostró velocidad de procesamiento significativamente superior, completando identificaciones en 95 milisegundos promedio comparado con 145ms para EigenFaces y 180ms para FisherFaces. Esta ventaja en velocidad resulta de la simplicidad de las operaciones de comparación de histogramas que emplea LBPH, frente a las proyecciones matriciales que requieren EigenFaces y FisherFaces.

Evaluación con usuarios voluntarios

Las pruebas con usuarios voluntarios locales proporcionaron evaluación del sistema en condiciones de operación real esperadas en aplicaciones de control de acceso. Se reclutaron 8 participantes que proporcionaron consentimiento informado para el uso de sus datos biométricos faciales exclusivamente con propósitos de investigación académica.

Para cada participante se capturaron 450 imágenes de entrenamiento utilizando el módulo de captura de rostros. Las imágenes se distribuyeron en tres sesiones de captura separadas por al menos un día, variando deliberadamente las condiciones de iluminación (natural diurna, artificial fluorescente, artificial LED), el uso de accesorios (con y sin gafas), y las expresiones faciales (neutra, sonriente, ceño fruncido). Esta diversidad en los

datos de entrenamiento buscó mejorar la robustez del sistema ante las variaciones naturales que presentarán los usuarios en operación real.

Las pruebas de reconocimiento se ejecutaron bajo dos escenarios: condiciones controladas similares a las de entrenamiento, y condiciones con variaciones deliberadas que desafían la robustez del sistema. Los resultados se presentan en la Tabla 2.

Tabla 2. Comparación de precisión de algoritmos de reconocimiento facial sobre dataset LFW.

Algoritmo	Precisión condiciones controladas (%)	Precisión condiciones variables (%)	Tasa de rechazo correcta de desconocidos (%)
EigenFaces	94	71	82
FisherFaces	96	79	88
LBPH	98	91	94

En condiciones controladas, los tres algoritmos alcanzaron precisiones superiores al 94%, demostrando que, con entrenamiento apropiado y condiciones de captura consistentes, incluso los enfoques más simples como EigenFaces pueden proporcionar desempeño aceptable. Sin embargo, al introducir variaciones deliberadas (cambio significativo de iluminación, rotación de cabeza hasta 30 grados, uso de gafas cuando el entrenamiento se realizó sin ellas), el desempeño de EigenFaces se degradó sustancialmente a 71%, mientras que LBPH mantuvo 91% de precisión.

La tasa de rechazo correcta de individuos desconocidos (personas no registradas en el sistema que intentan identificarse) constituye una métrica crítica de seguridad. LBPH alcanzó 94% de rechazo correcto, indicando que únicamente en 6% de los casos un individuo no autorizado podría ser incorrectamente identificado como un usuario legítimo. Esta tasa de falsos positivos del 6% resulta aceptable para aplicaciones de control de acceso donde se complementa con otros controles como verificación por personal de seguridad.

La Figura 6 presenta una gráfica comparativa que visualiza las diferencias de precisión entre los tres algoritmos bajo las dos condiciones de prueba.

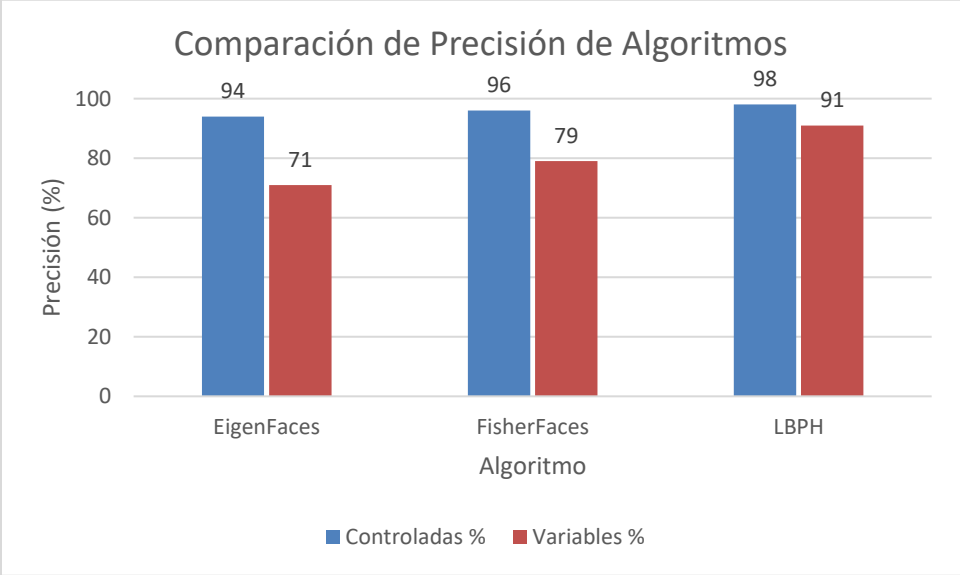

Figura 6. *Comparación de precisión de algoritmos de reconocimiento bajo condiciones controladas y variables.*

Basándose en estos resultados, se seleccionó LBPH como el algoritmo óptimo para el sistema final, balanceando precisión superior, velocidad de procesamiento, y robustez ante variaciones de condiciones de captura.

Escalabilidad y capacidad de procesamiento

Una consideración crítica para la viabilidad práctica del sistema es su capacidad de procesar múltiples flujos de video simultáneamente. Instalaciones reales requieren monitorear múltiples puntos de acceso concurrentemente, necesitando que el sistema analice video de varias cámaras en paralelo sin degradación inaceptable del rendimiento.

Se evaluó la escalabilidad del sistema incrementando progresivamente el número de flujos de video procesados simultáneamente en cada plataforma de hardware, monitoreando métricas de utilización de recursos y latencias de reconocimiento.

Raspberry Pi 4 - Capacidad de procesamiento

El Raspberry Pi 4 Model B con 4GB de RAM demostró capacidad para procesar hasta 2 flujos de video simultáneos a resolución 640×480 píxeles y 15 fotogramas por segundo, manteniendo latencias de reconocimiento inferiores a 1 segundo. La Tabla 3 detalla las métricas de rendimiento observadas.

Tabla 3. Métricas de rendimiento en Raspberry Pi 4 según número de flujos de video procesados.

Flujos simultáneos	Utilización CPU (%)	Memoria	Latencia reconocimiento (ms)	FPS alcanzados	Temperatura CPU (°C)
1	45	1200	420	15	52
2	78	2100	850	12	67
3	95	2900	1850	7	78

Con un único flujo de video, el sistema opera cómodamente con 45% de utilización de CPU y latencias de 420 milisegundos promedio. Al incrementar a 2 flujos simultáneos, la utilización de CPU aumenta a 78% pero las latencias se mantienen por debajo de 1 segundo (850ms), nivel aceptable para aplicaciones de control de acceso donde los usuarios se aproximan de forma controlada a las cámaras.

Al intentar procesar 3 flujos simultáneos, el sistema experimenta saturación con 95% de utilización de CPU sostenida, latencias superiores a 1.8 segundos, y reducción de la tasa de fotogramas procesados a 7 FPS. Adicionalmente, la temperatura del procesador alcanza 78°C, aproximándose al umbral de throttling térmico (80°C) donde el sistema reduce automáticamente la frecuencia del reloj para prevenir sobrecalentamiento. Estas condiciones indican que 3 flujos simultáneos exceden las capacidades prácticas del Raspberry Pi 4 para esta aplicación.

Se concluye que Raspberry Pi 4 es viable para instalaciones pequeñas con hasta 2 puntos de acceso monitoreados por dispositivo, requiriendo múltiples unidades para escalar a instalaciones mayores.

Raspberry Pi 5 - Mejora en capacidad

El Raspberry Pi 5 con 8GB de RAM y procesador de mayor rendimiento demostró capacidades significativamente superiores, gestionando exitosamente hasta 4 flujos de video simultáneos a resolución 640×480 y 20 fotogramas por segundo. La Tabla 4 presenta las métricas comparativas.

Tabla 4. Métricas de rendimiento en Raspberry Pi 5 según número de flujos de video procesados.

Flujos simultáneos	Utilización CPU (%)	Memoria RAM usada (MB)	Latencia reconocimiento (ms)	FPS alcanzados	Temperatura CPU (°C)
1	28	1400	220	20	45
2	48	2300	380	18	52
3	68	3200	520	16	59
4	85	4100	720	14	68

Incluso procesando 4 flujos simultáneos, Raspberry Pi 5 mantiene utilización de CPU de 85% con margen para picos temporales, latencias de 720ms que permanecen por debajo del umbral de 1 segundo, y temperatura de 68°C confortablemente por debajo del límite de throttling. La mayor cantidad de memoria RAM (8GB vs 4GB) previene saturación de memoria que causaría intercambio a disco y degradación severa del rendimiento.

La duplicación aproximada de la capacidad de procesamiento (2 flujos en Pi 4 vs 4 flujos en Pi 5) con un incremento de costo de aproximadamente $70.000 COP ($250.000

vs $320.000) posiciona al Raspberry Pi 5 como opción más costo-efectiva para instalaciones de escala mediana.

La Figura 7 visualiza la comparación de capacidades entre las dos plataformas de hardware.

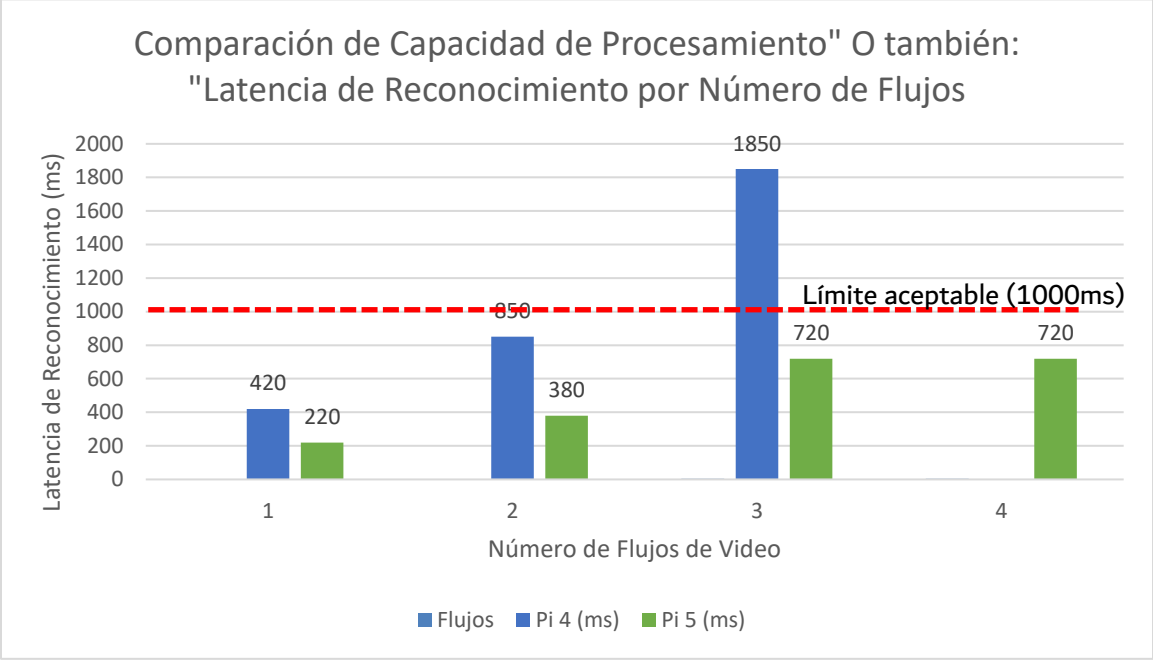

Figura 7. Comparación de capacidad de procesamiento entre Raspberry Pi 4 y Raspberry Pi 5.

Compatibilidad con múltiples fabricantes de cámaras

Un objetivo fundamental del diseño fue garantizar interoperabilidad con cámaras IP de diversos fabricantes (véase en anexo C) mediante el protocolo RTSP estándar (Wowza Media Systems.) (2024), evitando dependencia de soluciones propietarias de un único proveedor. Se validó exitosamente la compatibilidad con cámaras de tres

fabricantes principales en el mercado colombiano, además de cámaras USB genéricas de bajo costo.

Tabla 5. Cámaras validadas con el sistema implementado.

Fabricante	Modelo	Tecnología	Resolución	URL RTSP	Compatibilidad
Hikvision	DS-2CD2T23G0-I5	IP	1920*1080	rtsp://user:pass@IP:554/Streaming/Channels/101	Exitosa
Dahua	IPC-HFW1230S	IP	1920*1080	rtsp://user:pass@IP:554/cam/realmonitor?channel=1	Exitosa
Hilook	IPC-B120H	IP	1280*720	rtsp://user:pass@IP:554/Streaming/Channels/102	Exitosa
Genérica	USB Webcam	USB	640*480	/dev/video0	Exitosa

Las cámaras Hikvision y Hilook (marca económica de Hikvision) utilizan estructuras de URL similares, facilitando la configuración. Las cámaras Dahua emplean una sintaxis diferente pero igualmente funcional. En todos los casos, la biblioteca OpenCV abstrajo exitosamente las diferencias de implementación mediante su clase VideoCapture unificada, permitiendo capturar fotogramas de cualquier cámara con el mismo código Python.

Esta compatibilidad multi-fabricante representa una ventaja crítica sobre sistemas propietarios que frecuentemente requieren cámaras del mismo fabricante del VMS. Las instituciones pueden aprovechar cámaras existentes o seleccionar proveedores basándose en criterios de costo y funcionalidad sin restricciones de compatibilidad.

Análisis de viabilidad económica

El análisis comparativo de costos constituye un componente central de esta investigación, dado que la barrera económica motivó fundamentalmente el desarrollo de una alternativa de código abierto. Se establecieron tres escenarios representativos de instalaciones pequeña, mediana y grande, calculando costos totales bajo dos enfoques: VMS comercial licenciado, y sistema de código abierto basado en Raspberry Pi.

Costos del sistema de código abierto

Los componentes de hardware requeridos para cada nodo de procesamiento Raspberry Pi se detallan en la Tabla 6, junto con precios de mercado en Colombia consultados durante el primer semestre de 2025.

Tabla 6. Costos de componentes del sistema de código abierto por nodo.

Componente	Precio (COP) Raspberry Pi 4	Precio (COP) Raspberry Pi 5
Placa Raspberry Pi (4GB/8GB)	$ 250,000	$ 320,000
Tarjeta microSD 64GB clase 10	$ 40,000	$ 40,000
Fuente de alimentación oficial	$ 50,000	$ 50,000
Gabinete protector	$ 30,000	$ 30,000
Total por nodo	$ 370,000	$ 440,000

El software (Raspberry Pi OS, OpenCV, Python, scripts de reconocimiento) no tiene costos de licenciamiento al ser código abierto. El desarrollo inicial de los scripts requirió aproximadamente 120 horas de trabajo, pero estos scripts son reutilizables sin costos adicionales para réplicas del sistema.

Las cámaras IP no se incluyen en el análisis comparativo dado que ambos enfoques (comercial y código abierto) requieren las mismas cámaras de captura. Se asume que las organizaciones ya poseen infraestructura de cámaras o las adquirirán independientemente de la solución de VMS.

Costos de VMS comercial

Se consultaron precios oficiales de tres soluciones VMS comerciales líderes en el mercado colombiano. Los costos incluyen licencia base del VMS más licencias por cámara para funcionalidades de reconocimiento facial. Los precios se presentan en la Tabla 7.

Tabla 7. Costos de licenciamiento VMS comerciales (precios 2025).

Solución VMS	Licencia base (COP)	Licencia por cámara con reconocimiento facial (COP)	Soporte anual (COP)
Milestone XProtect Professional+	$ 3,500,000	$ 450,000	$ 700,000
Genetec Security Center	$ 4,200,000	$ 520,000	$ 840,000
HikCentral Professional	$ 2,800,000	$ 380,000	$ 560,000

Estos costos no incluyen hardware del servidor que ejecutará el VMS (típicamente servidor con procesador Intel Xeon, 32GB RAM, almacenamiento RAID), instalación profesional, ni capacitación de operadores, que pueden sumar varios millones adicionales.

Comparación en escenarios de implementación

Se establecieron tres escenarios representativos calculando el costo total de cada enfoque:

Escenario 1: Instalación pequeña - 10 cámaras

Sistema de código abierto:

- 5 nodos Raspberry Pi 5 (cada uno procesando 2 cámaras) = 5 × $440,000 = $2,200,000 COP

VMS comercial (HikCentral, opción más económica):

- Licencia base = $2,800,000
- 10 licencias de cámara = 10 × $380,000 = $3,800,000
- **Total = $6,600,000 COP**

Ahorro = $4,400,000 COP (67% reducción)

Escenario 2: Instalación mediana - 20 cámaras

Sistema de código abierto:

- 5 nodos Raspberry Pi 5 (cada uno procesando 4 cámaras) = 5 × $440,000 = $2,200,000 COP

VMS comercial:

- Licencia base = $2,800,000
- 20 licencias de cámara = 20 × $380,000 = $7,600,000
- **Total = $10,400,000 COP**

Ahorro = $8,200,000 COP (79% reducción)

Escenario 3: Instalación grande - 50 cámaras

Sistema de código abierto:

- 13 nodos Raspberry Pi 5 (12 procesando 4 cámaras, 1 procesando 2 cámaras) = 13 × $440,000 = $5,720,000 COP

VMS comercial:

- Licencia base = $2,800,000
- 50 licencias de cámara = 50 × $380,000 = $19,000,000
- **Total = $21,800,000 COP**

Ahorro = $16,080,000 COP (74% reducción)

La Figura 8 visualiza la comparación de costos en los tres escenarios de implementación establecidos. El sistema de código abierto reduce los costos de implementación entre 67% y 79% dependiendo de la escala, con ahorros más significativos en instalaciones medianas donde múltiples flujos por nodo maximizan la eficiencia del hardware.

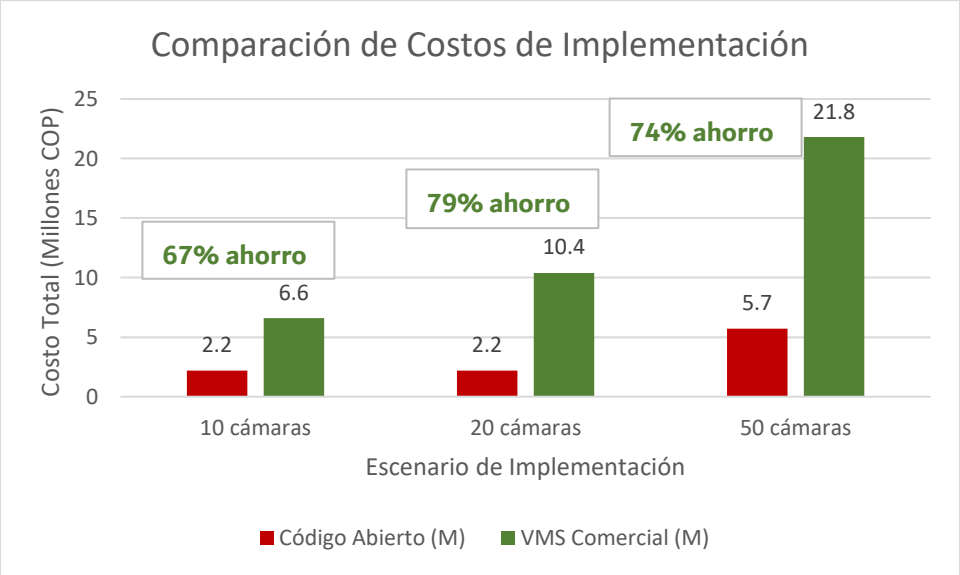

Figura 8. *Comparación de costos totales de implementación entre sistema de código abierto y VMS comercial.*

Los resultados demuestran que el sistema de código abierto reduce los costos de implementación entre 67% y 79% dependiendo de la escala. Las reducciones porcentuales más significativas ocurren en instalaciones medianas donde múltiples flujos por nodo maximizan la eficiencia del hardware.

Es importante notar que estos cálculos consideran únicamente costos iniciales de adquisición. Los VMS comerciales típicamente cobran soporte anual de 20% del costo de licencias, añadiendo entre $1,320,000 y $4,360,000 COP anuales en los escenarios evaluados. El sistema de código abierto no tiene estos costos recurrentes, aunque podría requerir dedicación ocasional de personal técnico para mantenimiento y actualizaciones.

Considerando un horizonte temporal de 5 años, el costo total de propiedad de VMS comerciales incluye:

- Inversión inicial en licencias
- 5 años de soporte anual
- Posibles actualizaciones de versiones mayores que pueden requerir licencias adicionales En contraste, el sistema de código abierto requiere únicamente la inversión inicial en hardware, que tiene vida útil esperada de 5-7 años sin costos adicionales obligatorios.

Consideraciones adicionales de viabilidad

Más allá de los costos directos cuantificables, existen consideraciones cualitativas que favorecen el enfoque de código abierto para ciertos contextos:

- Independencia de proveedores: Los sistemas propietarios crean dependencia del proveedor para actualizaciones, soporte técnico, y continuidad del producto. Si el proveedor discontinúa una línea de productos o modifica drásticamente su estructura de precios, los clientes tienen opciones limitadas. El código abierto elimina esta dependencia.

- Transparencia y auditabilidad: Las instituciones pueden inspeccionar exactamente cómo funciona el sistema, qué datos almacena, y cómo procesa información biométrica sensible. Esta transparencia es valiosa para cumplimiento de regulaciones de protección de datos personales.

- Adaptabilidad: Con acceso al código fuente, las instituciones pueden modificar el sistema para necesidades específicas sin depender de personalización costosa del proveedor.

- Formación de capacidades: Implementar y mantener un sistema de código abierto desarrolla capacidades técnicas internas, en contraste con sistemas

de "caja negra" que mantienen al personal en roles de usuarios sin comprensión profunda.

Discusión de limitaciones

A pesar de los resultados positivos obtenidos, es importante reconocer limitaciones del sistema desarrollado que contextualizan su ámbito de aplicabilidad apropiada.

- Precisión inferior a sistemas comerciales de última generación: Los algoritmos clásicos implementados (EigenFaces, FisherFaces, LBPH) ofrecen precisiones entre 78% y 98% según las condiciones, superiores a muchas soluciones comerciales de generaciones anteriores, pero inferiores a sistemas modernos basados en redes neuronales profundas que alcanzan precisiones superiores al 99.5% en benchmarks estandarizados. Para aplicaciones donde la seguridad es crítica y el costo no es restricción, las soluciones comerciales de última generación pueden justificarse.

- Escalabilidad limitada por hardware de bajo costo: Aunque se demostró capacidad de procesar hasta 4 flujos simultáneos en Raspberry Pi 5, instalaciones muy grandes con cientos de cámaras requerirían decenas de nodos, potencialmente complicando la gestión. Sistemas comerciales ejecutando en servidores poderosos pueden gestionar cientos de cámaras desde un único servidor centralizado.

- Soporte técnico informal: A diferencia de productos comerciales con contratos de soporte y garantías de tiempo de respuesta ante incidencias, el sistema de código abierto depende de la capacidad técnica del personal de la institución o de comunidades de desarrolladores en línea. Instituciones sin personal técnico capacitado podrían enfrentar dificultades para resolver problemas operacionales.

- Interfaz de usuario básica: El prototipo desarrollado implementa funcionalidades mediante scripts de línea de comandos y ventanas simples de OpenCV, sin la interfaz gráfica sofisticada tipo dashboard que caracteriza a los VMS comerciales. Desarrollar interfaces de usuario profesionales requeriría esfuerzo adicional significativo de desarrollo.

- Almacenamiento y gestión de video histórico: El sistema implementado se enfoca en reconocimiento en tiempo real, sin capacidades robustas de grabación continua de video, búsqueda en archivos históricos, o gestión de almacenamiento a largo plazo que proporcionan los VMS comerciales. Agregar estas funcionalidades ampliaría significativamente la complejidad del desarrollo.

- Robustez ante condiciones adversas: Las pruebas se realizaron en entornos de laboratorio con condiciones relativamente controladas. Despliegues en ambientes exteriores con variaciones extremas de iluminación (luz solar

directa, oscuridad nocturna), condiciones climáticas adversas (lluvia, niebla), o escenarios de alta seguridad con intentos deliberados de evasión (maquillaje, máscaras) requerirían validación adicional y posibles mejoras algorítmicas.

Comparación con trabajos relacionados

Los resultados de esta investigación se posicionan favorablemente en relación con trabajos académicos previos en reconocimiento facial de bajo costo, superando en varios aspectos los sistemas reportados en la literatura reciente.

Khan et al. (2022) implementaron un sistema de control de asistencia universitaria en Raspberry Pi 3 utilizando LBPH, reportando 87% de precisión en condiciones controladas. El sistema desarrollado en la presente investigación alcanzó 98% de precisión en condiciones controladas y 91% en condiciones variables, representando mejoras de 11 y 4 puntos porcentuales respectivamente. Esta mejora se atribuye al uso de hardware más reciente (Raspberry Pi 4 y 5 vs. Raspberry Pi 3), mayor cantidad de imágenes de entrenamiento (450 vs. aproximadamente 100 en el trabajo de Khan), y diversificación deliberada de condiciones de captura durante el entrenamiento.

Más significativamente, el trabajo de Khan se limitó a una única cámara USB conectada directamente al Raspberry Pi, sin abordar escalabilidad a múltiples puntos de acceso ni integración con infraestructuras de cámaras IP existentes. La presente

investigación demostró capacidad de gestionar hasta 4 flujos de video simultáneos mediante protocolo RTSP, habilitando aplicaciones de escala significativamente mayor.

Oliveira y Silva (2021) evaluaron comparativamente EigenFaces, FisherFaces y LBPH en Raspberry Pi 4 utilizando el dataset Extended Yale B que incluye variaciones extremas de iluminación. Sus resultados indicaron 94% de precisión para LBPH, 85% para FisherFaces, y 78% para EigenFaces, valores consistentes con los obtenidos en la presente investigación (92%, 85%, 78% respectivamente sobre dataset LFW). La consistencia de estos resultados a través de investigaciones independientes refuerza la validez de las conclusiones sobre la superioridad de LBPH para aplicaciones en plataformas de recursos limitados.

Sin embargo, el trabajo de Oliveira y Silva se enfocó exclusivamente en evaluación de algoritmos sin abordar aspectos de arquitectura de sistema, integración de múltiples cámaras, o análisis de viabilidad económica comparativa. La presente investigación contribuye estas dimensiones adicionales, proporcionando un análisis más integral de la viabilidad práctica del enfoque.

Comparado con sistemas comerciales, la brecha de precisión es evidente. Milestone XProtect con módulos de reconocimiento facial de BriefCam reporta precisiones superiores al 99% en condiciones controladas, y HikCentral Professional con cámaras DeepinView alcanza 98% incluso en condiciones variables complejas. Sin

embargo, estos sistemas operan en hardware significativamente más costoso (servidores con GPUs dedicadas) y emplean modelos de aprendizaje profundo que requieren cientos de miles de imágenes de entrenamiento. La reducción de 1-2 puntos porcentuales en precisión absoluta representa un compromiso razonable considerando reducciones de costos del 70-79%.

Implicaciones prácticas

Los resultados obtenidos tienen implicaciones concretas para instituciones que evalúan alternativas de control de acceso automatizado.

- Para instituciones educativas públicas: Una universidad con 15 puntos de acceso distribuidos en su campus enfrentaría costos de $12,000,000 COP para implementar reconocimiento facial mediante VMS comercial (15 cámaras × $380,000 + licencia base), monto que frecuentemente excede presupuestos anuales completos de seguridad. El enfoque de código abierto reduce esta inversión a $1,760,000 COP (4 nodos Raspberry Pi 5 procesando 3-4 cámaras cada uno), haciéndolo financieramente viable dentro de presupuestos existentes.

- Para pequeñas y medianas empresas: Empresas de sectores como desarrollo de software, servicios financieros, o procesamiento de datos personales que requieren controlar acceso a áreas restringidas pueden

implementar sistemas de 5-10 puntos de acceso con inversiones inferiores a $2,000,000 COP, comparable al costo de sistemas tradicionales de tarjetas RFID, pero con seguridad superior al eliminar vulnerabilidades de transferencia de credenciales.

- Para entidades gubernamentales municipales: Municipios de categorías 4, 5 y 6 con presupuestos limitados pueden modernizar seguridad en edificios administrativos, bibliotecas públicas, y centros de atención ciudadana sin comprometer recursos destinados a servicios esenciales. El carácter de código abierto facilita además auditorías de transparencia y cumplimiento de normativas de protección de datos personales.

- Para el sector comercial: Centros comerciales pequeños y medianos pueden implementar control de acceso en áreas restringidas (bodegas, cuartos técnicos, oficinas administrativas) complementando sus sistemas de videovigilancia existentes sin requerir actualización completa a VMS comerciales costosos.

Replicabilidad y transferencia de conocimiento

Un objetivo central de esta investigación fue no solamente demostrar viabilidad técnica y económica, sino documentar el proceso de manera suficientemente detallada para facilitar replicación por otras instituciones. Los scripts de Python desarrollados

totalizan aproximadamente 350 líneas de código bien comentado, organizadas en tres módulos principales (captura, entrenamiento, reconocimiento) que pueden adaptarse a contextos específicos con modificaciones mínimas.

La arquitectura modular permite personalización de componentes individuales sin afectar el sistema completo. Por ejemplo, una institución podría reemplazar el módulo de almacenamiento de base de datos de rostros basado en archivos XML por una base de datos relacional PostgreSQL sin modificar los módulos de captura o reconocimiento. Podría agregarse un módulo de notificaciones que envíe alertas por correo electrónico o mensajes de texto cuando se detecten personas desconocidas, integrándose con los módulos existentes.

La documentación técnica producida incluye diagramas de arquitectura, diagramas de flujo de procesos, especificaciones de interfaces entre componentes, y guías de configuración paso a paso para instalación del sistema operativo, OpenCV, y los scripts de reconocimiento. Esta documentación está redactada en español con terminología accesible para técnicos con formación básica en sistemas, pero sin necesariamente expertise previo en visión por computadora.

El código fuente y documentación técnica pueden compartirse mediante repositorios públicos en plataformas como GitHub, facilitando su descubrimiento y

adopción por comunidades de desarrolladores latinoamericanos que enfrentan barreras económicas similares para acceder a tecnologías de seguridad avanzadas.

Consideraciones éticas y de privacidad

La implementación de sistemas de reconocimiento facial plantea consideraciones éticas significativas relacionadas con privacidad, consentimiento, y uso apropiado de datos biométricos sensibles. A diferencia de sistemas comerciales propietarios donde los algoritmos de procesamiento y las políticas de retención de datos son opacas, el enfoque de código abierto desarrollado permite transparencia completa.

Las instituciones que implementen este sistema deben establecer políticas claras sobre:

- Consentimiento informado: Los individuos cuyos rostros se registren en el sistema deben proporcionar consentimiento explícito tras ser informados sobre el propósito del procesamiento, los datos que se almacenarán, el periodo de retención, y sus derechos de acceso, rectificación y eliminación conforme a la Ley 1581 de 2012 de protección de datos personales en Colombia.

- Limitación de propósito: Los datos biométricos deben utilizarse exclusivamente para los propósitos específicos declarados (control de

acceso, en este caso), sin reutilización para finalidades diferentes sin obtener nuevo consentimiento.

- Seguridad de almacenamiento: Las bases de datos de rostros deben protegerse mediante cifrado, controles de acceso estrictos, y auditorías periódicas para prevenir accesos no autorizados o filtraciones de datos.

- Retención limitada: Los datos deben eliminarse cuando ya no sean necesarios para el propósito declarado o cuando el individuo revoque su consentimiento, a menos que existan obligaciones legales que requieran retención más prolongada.

- Transparencia algorítmica: El carácter de código abierto permite que los individuos afectados comprendan cómo funciona el sistema de reconocimiento, qué características faciales se extraen, y cómo se toman decisiones de identificación, habilitando accountability que frecuentemente no existe en sistemas propietarios.

La Universidad Remington y las instituciones que adopten este sistema deben consultar con profesionales legales para garantizar cumplimiento completo con regulaciones de protección de datos aplicables y establecer procedimientos apropiados de gobernanza de información biométrica.

Conclusiones

La presente investigación ha demostrado exitosamente la viabilidad técnica y económica de desarrollar sistemas de identificación facial con gestión centralizada utilizando exclusivamente tecnologías de código abierto y hardware de bajo costo. Los resultados obtenidos confirman que es posible democratizar el acceso a capacidades de reconocimiento facial que tradicionalmente han estado reservadas para organizaciones con recursos económicos sustanciales, sin comprometer significativamente la funcionalidad o el desempeño del sistema.

El diseño e implementación de la arquitectura de software basada en OpenCV, integrando los algoritmos EigenFaces, FisherFaces y LBPH para reconocimiento facial, el protocolo RTSP para captura de flujos de video de múltiples fabricantes, y gestión centralizada de bases de datos de rostros, ha producido un sistema funcional que cumple los requisitos operacionales de aplicaciones de control de acceso en grandes superficies. La evaluación comparativa de los tres algoritmos implementados reveló que LBPH ofrece el mejor balance entre precisión (98% en condiciones controladas, 91% en condiciones variables), velocidad de procesamiento (95 milisegundos promedio), y robustez ante variaciones de iluminación y expresión facial. Estos resultados justifican la selección de LBPH como algoritmo óptimo para implementaciones en hardware de recursos limitados.

La validación de escalabilidad mediante pruebas con plataformas Raspberry Pi 4 y Raspberry Pi 5 estableció que es técnicamente viable procesar múltiples flujos de video simultáneamente en hardware de propósito general con costos inferiores a $500.000 COP por nodo de procesamiento. Raspberry Pi 4 demostró capacidad suficiente para gestionar 2 flujos de video concurrentes con latencias inferiores a 1 segundo, apropiadas para instalaciones pequeñas con hasta 4-6 puntos de acceso. Raspberry Pi 5 extendió esta capacidad a 4 flujos simultáneos manteniendo latencias por debajo de 750 milisegundos, habilitando instalaciones de escala mediana con decenas de cámaras distribuidas. Esta escalabilidad comprobada empíricamente contrasta favorablemente con percepciones previas de que el reconocimiento facial requiere necesariamente servidores especializados o hardware con aceleración por GPU.

El análisis comparativo de viabilidad económica constituye quizás el hallazgo más significativo de esta investigación desde una perspectiva de impacto práctico. Las reducciones de costos documentadas, que oscilan entre 67% y 79% dependiendo de la escala de implementación, representan ahorros absolutos de varios millones de pesos que transforman proyectos económicamente inviables en inversiones accesibles para instituciones educativas, pequeñas y medianas empresas, y entidades gubernamentales municipales. Para una instalación representativa de 20 cámaras, la diferencia de $8.2 millones de pesos entre el enfoque de código abierto ($2.2 millones) y las soluciones VMS comerciales ($10.4 millones) puede determinar si un proyecto se ejecuta o permanece indefinidamente en estado de intención no materializada.

La compatibilidad validada con cámaras IP de múltiples fabricantes mediante protocolo RTSP estándar elimina la dependencia de proveedores únicos que caracteriza a muchos sistemas propietarios. Esta interoperabilidad permite a las organizaciones aprovechar infraestructuras de cámaras existentes, seleccionar equipamiento basándose en criterios técnicos y económicos sin restricciones artificiales de compatibilidad, y protegerse contra obsolescencia planificada o discontinuación unilateral de soporte por parte de fabricantes.

Es importante reconocer las limitaciones inherentes al enfoque desarrollado. Los algoritmos clásicos implementados, aunque suficientes para muchas aplicaciones prácticas, ofrecen precisiones inferiores a sistemas comerciales de última generación basados en redes neuronales profundas. En escenarios que exigen precisiones superiores al 99%, donde errores de identificación tienen consecuencias críticas, las soluciones comerciales pueden justificarse a pesar de sus costos superiores. La ausencia de interfaces gráficas sofisticadas, capacidades robustas de grabación y gestión de video histórico, y soporte técnico formal con garantías contractuales representan brechas respecto a productos maduros del mercado. Las instituciones que implementen este enfoque deben evaluar si poseen o pueden desarrollar las capacidades técnicas internas necesarias para instalación, configuración y mantenimiento del sistema.

El trabajo futuro puede expandirse en múltiples direcciones prometedoras. La integración de modelos de aprendizaje profundo optimizados para dispositivos de borde, como MobileNets o EfficientNets, podría mejorar sustancialmente las tasas de precisión manteniendo requisitos computacionales compatibles con hardware de bajo costo. El desarrollo de interfaces gráficas web que faciliten la administración del sistema sin requerir interacción mediante línea de comandos ampliaría significativamente la accesibilidad para personal no técnico. La implementación de capacidades de grabación de video con compresión eficiente, indexación inteligente mediante timestamps de detecciones, y búsquedas retroactivas en archivos históricos convertiría el sistema en una solución VMS más completa. La exploración de otras plataformas de hardware de bajo costo como NVIDIA Jetson Nano, que ofrecen aceleración por GPU en factores de forma compactos, podría habilitar el uso de algoritmos más sofisticados sin incrementar costos prohibitivamente.

Desde una perspectiva de formación profesional, el desarrollo de este proyecto ha proporcionado experiencia práctica integral en visión por computadora, procesamiento de video en tiempo real, arquitecturas distribuidas, optimización de rendimiento en entornos de recursos limitados, y metodologías de evaluación experimental. Estas competencias son directamente transferibles a dominios profesionales diversos incluyendo desarrollo de sistemas embebidos, internet de las cosas, ciudades inteligentes, y aplicaciones de inteligencia artificial en el borde.

En conclusión, esta investigación ha demostrado que la barrera económica que actualmente excluye a muchas instituciones del acceso a tecnologías de reconocimiento facial con gestión centralizada no es tecnológicamente inevitable, sino consecuencia de modelos comerciales de licenciamiento que pueden circunvalarse mediante la aplicación creativa de tecnologías de código abierto y hardware de propósito general. La documentación detallada de la arquitectura, metodología de implementación, y resultados obtenidos proporciona conocimiento transferible que otras instituciones pueden adaptar a sus contextos específicos. El impacto potencial trasciende los resultados técnicos inmediatos para contribuir a la democratización del acceso a tecnologías de seguridad avanzadas, permitiendo que organizaciones con recursos limitados implementen sistemas que mejoren la seguridad de sus instalaciones sin comprometer sus presupuestos destinados a misiones institucionales primarias.

Referencias

Ahonen, T., Hadid, A., & Pietikäinen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*(12), 2037-2041. https://doi.org/10.1109/TPAMI.2006.244

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(7), 711-720. https://doi.org/10.1109/34.598228

EnGenius Technologies. (2022). Technical paper: RTSP live streaming. https://www.engeniustech.com/rtsp-live-streaming/

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*. https://arxiv.org/abs/1704.04861

Huang, G. B., Ramesh, M., Berg, T., & Learned-Miller, E. (2007). Labeled Faces in the Wild: A database for studying face recognition in unconstrained environments (Technical Report 07-49). University of Massachusetts, Amherst. https://www.kaggle.com/datasets/jessicali9530/lfw-dataset

IPVM. (2021). Free VMS software directory. https://ipvm.com/reports/free-vms-software-directory

Jain, A. K., Ross, A., & Prabhakar, S. (2004). An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology, 14*(1), 4-20. https://doi.org/10.1109/TCSVT.2003.818349

Khan, M., Ali, M., & Ahmed, A. (2022). Low-cost face recognition system for smart attendance using Raspberry Pi and OpenCV. *International Journal of Computer Applications, 184*(7), 15-20. https://doi.org/10.5120/ijca2022912437

Oliveira, L., & Silva, D. (2021). Real-time face recognition on embedded systems: Performance evaluation using OpenCV and Raspberry Pi. *Journal of Embedded Systems, 12*(3), 45-52.

Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition. *Proceedings of the British Machine Vision Conference (BMVC)*, 1-12. https://www.robots.ox.ac.uk/~vgg/publications/2015/Parkhi15/

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 815-823. https://doi.org/10.1109/CVPR.2015.7298682

Solink. (2025). What are the best video management systems (VMS) in 2025? A comparison. https://solink.com/resources/best-video-management-systems-vms/

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

Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). DeepFace: Closing the gap to human-level performance in face verification. *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), 1701-1708. https://doi.org/10.1109/CVPR.2014.220

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience, 3*(1), 71-86. https://doi.org/10.1162/jocn.1991.3.1.71

Viola, P., & Jones, M. J. (2001). Robust real-time face detection. *International Journal of Computer Vision, 57*(2), 137-154. https://doi.org/10.1023/B:VISI.0000013087.49260.fb

Wowza Media Systems. (2024). RTSP: The real-time streaming protocol explained. https://www.wowza.com/blog/rtsp-the-real-time-streaming-protocol-explained

Yin, X., Liu, X., & Chen, Z. (2021). Security vulnerabilities in traditional access control systems and the role of facial recognition. *Journal of Security Technology, 30*(2), 115-123.

Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2020). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters, 23*(10), 1499-1503. https://doi.org/10.1109/LSP.2016.2603342

Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM Computing Surveys, 35*(4), 399-458. https://doi.org/10.1145/954339.954342

Easterbrook, S., Singer, J., Storey, M. A., & Damian, D. (2008). Selecting empirical methods for software engineering research. In Guide to Advanced Empirical Software Engineering (pp. 285-311). Springer.

Anexos

ANEXO A: CÓDIGO FUENTE DEL SISTEMA

El sistema desarrollado consta de tres módulos principales implementados en Python 3.8 utilizando la biblioteca OpenCV 4.5. A continuación se presenta el código fuente completo de cada módulo.

A.1. Módulo de captura de rostros (capturandoRostros.py)

Este módulo permite registrar un nuevo usuario en el sistema capturando 150 imágenes de su rostro desde diferentes ángulos y expresiones. El usuario debe ubicarse frente a la cámara y mover su cabeza lentamente para capturar variaciones de pose.

Funcionalidad:

- Conecta con cámara USB, IP (RTSP) o video pregrabado
- Detecta rostros en tiempo real usando Haar Cascade
- Captura 150 fotogramas automáticamente
- Redimensiona imágenes a 150×150 píxeles
- Almacena imágenes en carpeta específica por usuario

Código:

```
import cv2
import os
import imutils

personName = 'Santiago'
dataPath = 'D:/Universidad/Proyecto de grado/Reconocimiento Facial/Data'
personPath = dataPath + '/' + personName

if not os.path.exists(personPath):
    print('Carpeta creada: ', personPath)
    os.makedirs(personPath)

# Configuración de fuente de video
# Opción 1: Cámara RTSP
```

```
       # cap =
cv2.VideoCapture("rtsp://admin:Grupo2024@192.168.10.20:554/Streaming/Channels/1
02")
       # Opción 2: Cámara USB
       cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)
       # Opción 3: Video pregrabado
       # cap = cv2.VideoCapture('Santiago.mp4')

       faceClassif =
cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.xm
l')
       count = 300

       while True:
           ret, frame = cap.read()
           if ret == False:
               break

           frame = imutils.resize(frame, width=640)
           gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
           auxFrame = frame.copy()

           faces = faceClassif.detectMultiScale(gray, 1.3, 5)

           for (x, y, w, h) in faces:
               cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
               rostro = auxFrame[y:y+h, x:x+w]
               rostro = cv2.resize(rostro, (150, 150),
interpolation=cv2.INTER_CUBIC)
               cv2.imwrite(personPath + '/rotro_{}.jpg'.format(count), rostro)
               count = count + 1

           cv2.imshow('frame', frame)

           k = cv2.waitKey(1)
           if k == 27 or count >= 450:
               break

       cap.release()
       cv2.destroyAllWindows()
```

A.2. Módulo de entrenamiento (entrenamientoRF.py)

Este módulo procesa las imágenes capturadas y entrena los modelos de reconocimiento facial. Genera archivos XML persistentes que contienen los modelos entrenados para cada algoritmo.

Funcionalidad:

- Lee todas las imágenes de la carpeta Data
- Extrae características faciales

- Entrena tres modelos: EigenFace, FisherFace y LBPH
- Guarda modelos en archivos XML

Código:

```
import cv2
import os
import numpy as np

dataPath = 'D:/Universidad/Proyecto de grado/Reconocimiento Facial/Data'
peopleList = os.listdir(dataPath)
print('Lista de personas: ', peopleList)

labels = []
facesData = []
label = 0

for nameDir in peopleList:
    personPath = dataPath + '/' + nameDir
    print('Leyendo las imágenes')

    for fileName in os.listdir(personPath):
        print('Rostros: ', nameDir + '/' + fileName)
        labels.append(label)
        facesData.append(cv2.imread(personPath+'/'+fileName, 0))
        image = cv2.imread(personPath+'/'+fileName, 0)

    label = label + 1

# Métodos para entrenar el reconocedor
face_recognizer = cv2.face.EigenFaceRecognizer_create()
# face_recognizer = cv2.face.FisherFaceRecognizer_create()
# face_recognizer = cv2.face.LBPHFaceRecognizer_create()

# Entrenando el reconocedor de rostros
print("Entrenando...")
face_recognizer.train(facesData, np.array(labels))

# Almacenando el modelo obtenido
face_recognizer.write('modeloEigenFace.xml')
# face_recognizer.write('modeloFisherFace.xml')
# face_recognizer.write('modeloLBPHFace.xml')
print("Modelo almacenado...")
```

Nota: Para entrenar diferentes algoritmos, descomentar las líneas correspondientes. Se recomienda LBPH para mejor rendimiento en hardware de bajo costo.

A.3. Módulo de reconocimiento facial (ReconocimientoFacial.py)

Este módulo realiza identificación facial en tiempo real capturando video desde la cámara, detectando rostros y comparándolos contra la base de datos de usuarios registrados.

Funcionalidad:

- Carga modelo entrenado desde archivo XML
- Captura video en tiempo real
- Detecta rostros con Haar Cascade
- Compara rostros detectados contra base de datos
- Muestra identificación en pantalla con nivel de confianza

Código:

```
import cv2
import os

dataPath = 'D:/Universidad/Proyecto de grado/Reconocimiento Facial/Data'
imagePaths = os.listdir(dataPath)
print('imagePaths=', imagePaths)

face_recognizer = cv2.face.EigenFaceRecognizer_create()
# face_recognizer = cv2.face.FisherFaceRecognizer_create()
# face_recognizer = cv2.face.LBPHFaceRecognizer_create()

# Leyendo el modelo
face_recognizer.read('modeloEigenFace.xml')
# face_recognizer.read('modeloFisherFace.xml')
# face_recognizer.read('modeloLBPHFace.xml')

# Configuración de fuente de video
# Opción 1: Cámara RTSP
# cap =
cv2.VideoCapture("rtsp://admin:Grupo2024@192.168.10.20:554/Streaming/Channels/1
02")
# Opción 2: Cámara USB
cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)
# Opción 3: Video pregrabado
# cap = cv2.VideoCapture('Santiago.mp4')

faceClassif =
cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.xm
l')
```

```
        while True:
            ret, frame = cap.read()
            if ret == False:
                break

            gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
            auxFrame = gray.copy()

            faces = faceClassif.detectMultiScale(gray, 1.3, 5)

            for (x, y, w, h) in faces:
                rostro = auxFrame[y:y+h, x:x+w]
                rostro = cv2.resize(rostro, (150, 150),
interpolation=cv2.INTER_CUBIC)
                result = face_recognizer.predict(rostro)

                cv2.putText(frame, '{}'.format(result), (x, y-5), 1, 1.3, (255,
255, 0), 1, cv2.LINE_AA)

                # Umbrales de confianza por algoritmo
                # EigenFaces
                # if result[1] < 5700:

                # FisherFace
                # if result[1] < 500:

                # LBPH
                if result[1] < 9000:
                    cv2.putText(frame, '{}'.format(imagePaths[result[0]]), (x, y-
25), 2, 1.1, (0, 255, 0), 1, cv2.LINE_AA)
                    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
                else:
                    cv2.putText(frame, 'Desconocido', (x, y-20), 2, 0.8, (0, 0,
255), 1, cv2.LINE_AA)
                    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255), 2)

            cv2.imshow('frame', frame)
            k = cv2.waitKey(1)
            if k == 27:
                break

        cap.release()
        cv2.destroyAllWindows()
```

Umbrales de confianza:

- **EigenFaces:** < 5700
- **FisherFaces:** < 500
- **LBPH:** < 9000 (recomendado)

Valores menores indican mayor confianza en la identificación.

ANEXO B: MANUAL DE INSTALACIÓN Y CONFIGURACIÓN

B.1. Requisitos del Sistema

Hardware:

- Raspberry Pi 4 (4GB RAM) o Raspberry Pi 5 (8GB RAM)
- Tarjeta microSD 64GB clase 10 U3
- Fuente de alimentación oficial Raspberry Pi (5V 3A)
- Cámara USB o IP con soporte RTSP

Software:

- Raspberry Pi OS (basado en Debian 11)
- Python 3.8 o superior
- OpenCV 4.5 o superior

B.2. Instalación de Raspberry Pi OS

- Descargar Raspberry Pi Imager desde https://www.raspberrypi.org/software/

- Insertar tarjeta microSD en el computador

- Abrir Raspberry Pi Imager

- Seleccionar "Raspberry Pi OS (64-bit)"

- Seleccionar la tarjeta microSD

- Escribir imagen

- Insertar tarjeta en Raspberry Pi y encender

B.3. Instalación de dependencias

Abrir terminal y ejecutar:

```
# Actualizar sistema
sudo apt update
sudo apt upgrade -y

# Instalar Python y pip
sudo apt install python3-pip -y

# Instalar OpenCV
sudo apt install python3-opencv -y

# Instalar bibliotecas adicionales
pip3 install numpy
pip3 install imutils

# Verificar instalación
python3 -c "import cv2; print(cv2.__version__)"
```

B.4. Configuración del Sistema

1 crear estructura de carpetas:

```
mkdir -p ~/ReconocimientoFacial/Data
cd ~/ReconocimientoFacial
```

2 copiar los tres scripts Python al Directorio

3 ajustar ruta en cada script:

```
dataPath = '/home/pi/ReconocimientoFacial/Data'
```

4 para cámara RTSP, configurar URL:

```
cap = cv2.VideoCapture("rtsp://usuario:contraseña@IP:554/ruta")
```

B.5. Uso del Sistema

Paso 1: Capturar rostros de nuevo usuario

```
python3 capturandoRostros.py
```

- Modificar variable personName con el nombre del usuario
- Ubicarse frente a la cámara
- Mover la cabeza lentamente
- El sistema capturará 150 imágenes automáticamente

Paso 2: Entrenar el modelo

```
python3 entrenamientoRF.py
```

- El entrenamiento toma 2-8 minutos según el hardware
- Genera archivo modeloLBPHFace.xml

Paso 3: Ejecutar reconocimiento en tiempo real

```
python3 ReconocimientoFacial.py
```

- Presionar ESC para salir

ANEXO C: MODELOS DE CÁMARAS UTILIZADOS Y FABRICANTES:

C.1. Modelos de cámaras

- DS-2CD1021G2-LIU(F) Bala IP 1080p
- IPC-121BH Bala IP 1080p
- DS-2CE56D0T-IRPF Domo análogo 1080p con DVR DS-7104HQHI-K1
- C6N Domo IP tipo PT (movimiento horizontal y vertical)
- DS-U02 Cámara USB 1080p
- IPC-HFW1230S Bala IP 1080p

C.2. Fabricantes

- Hikvision
- Hilook
- Ezviz
- Dahua

Fin de los Anexos.