



Aplicación de técnicas de inteligencia artificial para reconocimiento facial en sistemas de seguridad en ambientes de intranet

Artificial Intelligence techniques application for facial detection in intranet environmental security systems


 <https://doi.org/10.52948/mare.v4i1.682>

JORGE ARMANDO PORTAL DÍAZ

 <http://orcid.org/0000-0003-1360-4930>

Universidad Central "Marta Abreu" de Las Villas (UCLV), Cuba
jportal@uclv.edu.cu

IRINA B. SILES SILES

 <http://orcid.org/0000-0002-9478-2597>


Universidad Central "Marta Abreu" de Las Villas (UCLV), Cuba
irinass@uclv.edu.cu

EDUARDO ENRIQUE PUIG CONTRERAS

 <http://orcid.org/0000-0003-4848-9549>

Universidad Central "Marta Abreu" de Las Villas (UCLV), Cuba
epuig@uclv.edu.co

ANIA MARÍA SÁNCHEZ

 <http://orcid.org/0000-0003-2393-9590>

Universidad Central "Marta Abreu" de Las Villas (UCLV), Cuba
aniasanchez@nauta.co.cu

Artículo de investigación

Recepción: 18 de enero de 2022

Aceptación: 12 de septiembre de 2022

Cómo citar este artículo

J. Portal, I. Siles, E. Puig y A. Sánchez, "Aplicación de técnicas de inteligencia artificial para reconocimiento facial en sistemas de seguridad en ambientes de intranet," Mare Ingenii, vol. 4, n.º 2, oct. 2022.

Resumen:

El vertiginoso crecimiento y la precisión de las técnicas de Inteligencia Artificial (AI, del inglés *Artificial Intelligence*) permiten analizar grandes volúmenes de datos de forma rápida y eficiente. En ese sentido, la aplicación de técnicas de reconocimiento facial en sistemas de seguridad (video-vigilancia) no quedan exentas y resultan convenientes pues asistirían al desempeño humano en las labores de observación, interpretación y etiquetado de imágenes en tiempo real, a la vez que funcionarían como un sistema de alerta o alarma ante la presencia de intrusos. La implementación de estos sistemas puede llevarse a cabo con hardware relativamente barato y aprovechando la capacidad de procesamiento del clúster big data de la Universidad Central "Marta Abreu" de Las Villas (UCLV). Con la puesta en práctica del proyecto se ofrecen soluciones a las problemáticas identificadas en la dirección de informatización asociadas a la gestión de cuentas por parte de los usuarios y aplicaciones futuras relacionadas con la detección de personal en áreas de interés. Con la implementación se pretenden dos posibles contribuciones: en primer lugar, se ha de diseñar un procedimiento capaz de ensamblar un conjunto de datos a gran escala, minimizando al mismo tiempo la cantidad de anotaciones manuales involucradas. Este procedimiento se ha de desarrollar para caras, pero evidentemente es adecuado para otras clases de objetos, así como para tareas específicas. La segunda contribución ha de ser mostrar que una Red Neuronal Convolutiva (CNN, del inglés *Convolutional Neural Network*), profunda con la formación adecuada, puede lograr resultados comparables a los del estado de la técnica.

Palabras clave: inteligencia artificial; visión artificial; detección de caras.

Abstract:

The rapid growth and precision of Artificial Intelligence (AI) techniques enable large volumes of data to be analyzed quickly and efficiently. Following these premises, the application of facial recognition techniques in security systems (video-surveillance) are not exempt and are convenient because they would assist human performance in the tasks of observation, interpretation and labeling of images in real time, at the same time that would function as an alert or alarm system in the presence of intruders. The implementation of these systems can be conducted with cheap hardware and taking advantage of the processing capacity of the big data cluster of the Universidad Central "Marta Abreu" de Las Villas (UCLV). With the implementation, two contributions intended: first, a procedure has to be designed that is capable of assembling a large-scale data set, while minimizing the number of manual annotations involved. This procedure has been developed for faces, but it is obviously suitable for other classes of objects, as well as for specific tasks. The second contribution must be to show that a Convolutional Neural Network (CNN) deep with the appropriate training can achieve results comparable to those of the state of the art.

Keywords: Artificial Intelligence; computer vision; face detection.

Introducción

La inteligencia artificial (AI, del inglés *Artificial Intelligence*) es el presente. Ya ha quedado lejos su posición de tecnología del futuro. Cada vez son más las empresas y organizaciones que se suman al uso de la AI como herramienta para mejorar la eficiencia de sus diferentes procesos de negocio. La automatización, precisión y rapidez en el análisis de datos complejos son elementos clave que estos sistemas

dominan a la perfección [1], [2]. Según [3], es apreciable que la tecnología, cada vez más avanzada, pone a disposición de los investigadores nuevas herramientas capaces de alcanzar importantes objetivos, y estas herramientas son grandes puntos de partida en sí mismas. Uno de los campos específicos en los que la AI ha ganado un desarrollo inmenso, implicando un sinnúmero de aplicaciones, es en la visión por computador (CV, del inglés *computer vision*).

Actualmente se están realizando muchas investigaciones en el campo de la CV, pero no se trata solo de investigación. Las aplicaciones del mundo real demuestran lo importante que es la CV para las empresas, el entretenimiento, el transporte, la atención médica y la vida cotidiana [4]. En el sitio web *Papers with Code* [5] se pueden observar y explorar el estado de la técnica en ramas más específicas del campo de CV tales como la clasificación de imágenes, la generación de imágenes, la segmentación en el sector de la medicina y el reconocimiento facial (FR, del inglés *facial recognition*). Según [6], el reconocimiento facial es una tecnología popular e importante integrada en muchas aplicaciones como pagos, desbloqueo de puertas, sistemas de monitoreo de video, etc. Un sistema de FR es una tecnología que puede identificar y verificar personas a partir de grabaciones e imágenes digitales. Implica la generación de grandes cantidades de datos, referentes tanto a imágenes como a grabaciones, incluso a *streaming* de video en tiempo real. Un factor clave para el crecimiento de estas aplicaciones es la avalancha de información visual que fluye desde teléfonos inteligentes, sistemas de seguridad, cámaras de tráfico y otros dispositivos visualmente instrumentados. Estos datos podrían desempeñar un papel importante en las operaciones

de todas las industrias, pero hoy en día no se utilizan a cabalidad. La información crea un banco de pruebas para entrenar aplicaciones de visión por computadora y una plataforma de lanzamiento para que se conviertan en parte de una variedad de actividades humanas [4].

De otro lado, el big data se entiende como el conjunto de procedimientos computacionales aplicados para analizar gran cantidad de datos con el fin de extraer información que presente ciertos patrones, relaciones y asociaciones relevantes para una organización. Debido a que progresivamente se crea más información en diferentes campos del saber, la tendencia para los próximos años es la integración masiva y recurrente del big data con la Internet de las Cosas (IdC) en un entorno urbano e industrial, donde no solo se monitorearán variables como la calidad del aire, temperatura y humedad relativas, sino también la integración con sistemas biométricos, cámaras de monitoreo y vigilancia [7].

La sinergia entre las técnicas de AI, CV (específicamente FR) y big data puede dar al traste con un sistema capaz de realizar aportes importantes en sistemas de FR en tiempo real, como pueden ser la autenticación, control de asistencia y sistemas de videovigilancia. La anterior aseveración es demostrada en las investigaciones de [7].

El campus universitario no queda exento de esta problemática. En todos los ambientes empresariales se hace necesaria la identificación de personal no autorizado, la detección de personas, control de acceso de menores, detección de actividades, de emociones, etc. El presente trabajo pretende conformar una propuesta de sistema que de conjunto permita acumular, ensamblar y organizar el conjunto de datos de gran escala asociado a los usuarios (alumnos,

personal docente y no docente) y sus rostros, para minimizar el número de anotaciones manuales involucradas en la recopilación de la información necesaria. Además, se pretende demostrar que con el uso de CNN el sistema consigue resultados comparables con el estado actual de la técnica, haciendo uso de las posibilidades que brinda el clúster big data de la UCLV y adicionales recursos de bajo costo.

Metodología

La presente investigación pretende proponer un sistema de FR con el objetivo de efectuar reconocimiento de rostros en *streams* de video en tiempo real con la posibilidad de aplicarlo a futuro en diversos ámbitos, como videovigilancia, autenticación y control de asistencia. Para ello se ha tenido en cuenta la integración de varios componentes con el fin de proponer un sistema escalable, eficiente y preciso. Se procura elaborar un método de conformación automática del set de datos necesarios para los entrenamientos de los algoritmos de FR, su integración con un flujo de trabajo big data, el cual posibilite la corrida rápida de los algoritmos antes mencionados y la visualización de los resultados de estos.

A. Construcción del dataset para el reconocimiento facial

La creación de un dataset se aborda como un proceso tedioso y lento según [15] por lo que sería conveniente elaborar una solución automatizada para esta tarea. La propuesta reside en una aplicación web realizada con el framework de desarrollo Django [16] en su versión 2.2.5, el cual está basado en el lenguaje de programación Python [17]. Este framework ha sido elegido por la facilidad que implica su uso en la corrida de

scripts en Python, que es a su vez el lenguaje de programación escogido para desarrollar los algoritmos de FR necesarios para el sistema. Es importante destacar la sinergia que se crea también con OpenCV [18], librería de CV optimizada y enfocada en aplicaciones en tiempo real, puesto que esta es *open source* (de código abierto), posee una interfaz para Python y es multiplataforma [18].

Esta aplicación web se encargará de la introducción de los datos asociados a las personas que se añadirán al sistema con el propósito de reconocer sus rostros en videos en tiempo real. Para ello se cuenta con dos procesos: 1) la introducción de los datos asociados al carné de identidad de las personas y 2) la subida de un conjunto de fotos para el entrenamiento de los modelos.

B. Tecnologías y arquitecturas de big data

Entrenar los algoritmos de reconocimiento facial complejos en imágenes y videos puede llevar horas, días o incluso semanas. En la mayoría de los casos, una sola máquina multi-GPU es suficiente para entrenar modelos grandes en un período de tiempo razonable. Sin embargo, para cargas de trabajo de reconocimiento facial en tiempo real más exigentes, la distribución de cargas computacionales en múltiples máquinas puede reducir drásticamente el tiempo de entrenamiento, lo que permite una experimentación iterativa rápida y acelera las implementaciones de aprendizaje profundo. Por lo tanto, las arquitecturas de procesamiento de big data y los marcos de procesamiento en paralelo como MapReduce y Spark juegan un papel clave en tales fronteras.

Hay dos enfoques principales para mejorar el reconocimiento facial, a saber, el paralelismo de modelos y el paralelismo de datos. La arquitectura de big data

se basa en la computación en clústeres distribuidos para las tareas de preprocesamiento y clasificación de los flujos de big data heterogéneos que emanan de diversas fuentes de datos. En este contexto, el paralelismo de datos se puede lograr desde el stack regular de Hadoop MapReduce. Sin embargo, cuando se trata de paralelismo de modelos, los marcos de trabajo en memoria juegan un papel clave. Por lo tanto, se propone en esta investigación utilizar el procesamiento en memoria de Spark para lograr un modelo de aprendizaje paralelo.

El sistema de procesamiento de big data que se propone ha de ser lo suficientemente poderoso como para identificar o verificar un rostro a partir de imágenes y videos digitales, que generalmente denominamos reconocimiento facial. Este sistema funciona comparando los rasgos faciales más comunes y destacados de una imagen determinada (recopilada en tiempo real de varias fuentes externas) con los rostros almacenados en una base de datos. El sistema de reconocimiento facial también tiene la capacidad de comprender patrones y variaciones basados en las características faciales de un individuo para reconocer de manera única a una persona.

C. Sistema de reconocimiento facial

Como sistema informático, el FR permite efectuar la detección, el seguimiento y la identificación o verificación de rostros a partir de imágenes o videos capturados con cámaras digitales. El reconocimiento facial se basa en una identificación de patrones visuales. En [8], [9] los autores expresan que generalmente un sistema de reconocimiento facial se divide en cuatro fases: detección, preprocesado o alineación del rostro, extracción de características, así como comparación y clasificación. Lo anterior se refleja en la Fig. 1.

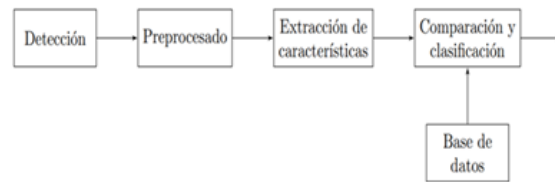


Fig. 1. Diagrama de un sistema de reconocimiento facial. [7]

La primera fase de detección facial consiste en localizar el rostro en la imagen. A continuación, se realiza un preprocesado a la imagen para alinear y normalizar los rostros, de manera que se obtengan unas características geométricas en común con todas las imágenes que se procesen. Luego se lleva a cabo una extracción de características faciales para obtener información útil que será usada para distinguir unas caras de otras. Finalmente, el vector de características obtenido será comparado con la base de datos para decidir a quién pertenece el rostro [10].

A partir de lo anterior se procede a la selección de los métodos de detección a emplearse, a la vez que se va conformando el conjunto de datos o dataset para separarlo en dos subconjuntos, uno para el entrenamiento y otro para probar y evaluar el sistema. Posteriormente se escogen y programan los algoritmos necesarios para las etapas de detección de los rostros, la extracción de las características faciales y la comparación y clasificación, los cuales son aplicados a los datos de entrenamiento.

La detección de rostros es el primer paso en el reconocimiento facial automatizado. Su confiabilidad tiene una gran influencia en el rendimiento y la usabilidad de todo el sistema de reconocimiento facial. Dada una sola imagen o un video, un detector de rostros ideal debería poder identificar y localizar todos los rostros presentes, independientemente de su

posición, escala, orientación, edad y expresión. Además, la detección debe ser independiente de condiciones de iluminación extrañas y del contenido de imagen y video [10]. Según [11] la detección de rostros en imágenes involucra cuatro métodos: *knowledge based*, *facial invariant*, *template matching* y *appearance based*. Viola y Jones en [12] exponen un método para efectuar la detección facial de manera rápida y en hardware barato, el cual ha sido la base para muchas aplicaciones en este sentido. En la actualidad se presentan distintas soluciones más precisas y avanzadas como lo son el uso de CNN, DNN [13], Histogramas de Gradientes Orientados o HOG (del inglés, *Histogram of Oriented Gradients*), SVM [14], entre otras. La eficiencia demostrada por HOG en los estudios de [1] indica este método como ideal para ser usado en esta investigación.

Para realizar el ajuste de las imágenes y garantizar en mejor medida la correcta interpretación de las variaciones de pose y escala se utiliza el algoritmo de estimación del punto de referencia, implementado en la librería *face_recognition* y basado en los estudios de V. Kazemi y J. Sullivan [15]. La idea básica es que se crean 68 puntos específicos (llamados *face landmarks*) que existen en cada cara: la parte superior del mentón, el borde exterior de cada ojo, el borde interior de cada ceja, etc. La librería *dlib* aporta el estimador ya entrenado para encontrar los puntos faciales según [16].

Después del ajuste y normalización de los rostros, la extracción de características es realizada para proveer información efectiva que será útil para distinguir entre personas distintas [10]. El algoritmo elegido para desempeñar esta tarea es una red convolucional que funciona bajo la función de pérdida del triplete (Fig. 2), la cual compara el rostro de una persona con una foto de la mis-

ma persona y la de otra distinta. Allí modifica los pesos de la red neuronal a lo largo de las iteraciones para hacer que las medidas de las fotos pertenecientes a la misma persona estén más cercanas y las que pertenecen a personas distintas se alejen.

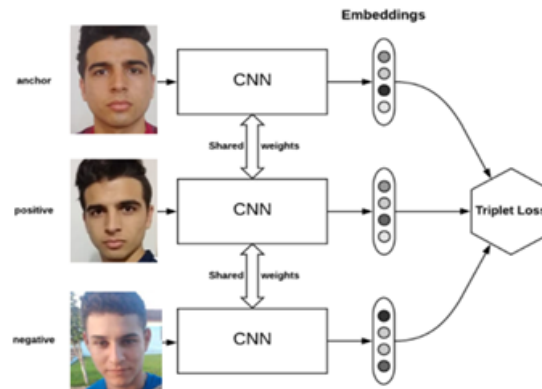


Fig. 2. Red convolucional funcionando bajo la función de pérdida del triplete.

Como último paso se procede a la comparación y clasificación de las representaciones obtenidas para cada rostro; se realiza una evaluación de tales representaciones de manera que se pueda comparar con las entradas existentes en las bases de datos y lograr una clasificación si esta concuerda con alguna de la base de datos o si se trata de un rostro desconocido. Se propone el uso de las técnicas de K-vecinos más cercanos (KNN, del inglés *K-Nearest Neighbors*) y máquinas de vectores de soporte (SVM, del inglés *support vector machines*).

Resultados y discusión

Los resultados obtenidos se presentan desde tres aristas diferentes aunque el producto final sea la implementación de un procedimiento que sea capaz de ensamblar un conjunto de datos a gran escala, minimizando al mismo tiempo la cantidad de anotaciones manuales involucradas para la FR en el ámbito

universitario. En consecuencia, se logra la obtención de una interfaz de usuario para la creación de set de datos de manera automática; se realiza una propuesta de una arquitectura de big data (se implementa de manera parcial) y se implementa el entrenamiento de la red de RF a pequeña escala de usuarios.

A. Creación del dataset automatizado

A partir de las necesidades de disminuir las anotaciones y búsquedas manuales de fotos se desarrolló una solución conveniente al problema presentado. Los procesos de introducción de personas y fotos de sus rostros al sistema pueden realizarse de forma manual, digitando los datos y escogiendo un grupo de imágenes a subir disponibles en una PC, tablet o smartphone. Lo anterior no supondría ninguna ventaja, por lo tanto, es recomendable utilizar las opciones automáticas que se ofrecen (Fig. 3).



Fig. 3. Interfaz de usuario para añadir datos de las personas.

Para ejecutar la adición automática de información de las personas se utiliza el código QR del carnet de identidad como fuente. El sistema corre un script de Python con funcionalidades de OpenCV que permite la detección y decodificación del QR presente en la imagen capturada por la webcam mediante la librería Pyzbar, para añadir a la base de datos una entrada con los nombres y apellidos de la persona en cuestión y su número de identidad.

La creación del conjunto de fotos individual de cada persona, ocurre de ma-

nera similar. Por comandos de Python es encendida la webcam para capturar una secuencia de imágenes que contengan el rostro de la persona. Hasta que no se hayan tomado el número de fotos establecido en el código, la cámara continuará encendida y analizando cada *frame* en busca de un rostro presente (Fig. 4a). El detector utilizado en esta parte del proceso es un detector *Haar cascade*. Este tipo de detectores es posiblemente el algoritmo de detección de objetos más popular de OpenCV [19], y fue presentado por primera vez por Paul Viola y Michael Jones en su publicación *Rapid Object Detection Using a Boosted Cascade of Simple Features* [20].

Existen muchos algoritmos que son más precisos que las cascadas de Haar (HOG + SVM lineal, SSD, Faster *R-CNN*, *YOLO*, por nombrar algunos), pero estas siguen siendo relevantes y útiles en la actualidad. Uno de los principales beneficios de las cascadas de Haar es que son tan rápidas que es difícil superar su velocidad. La desventaja que presentan es que tienden a ser propensas a detecciones de falsos positivos; por tanto, requieren un ajuste de parámetros cuando se aplican para inferencia o detección y, en general, no son tan precisas como los algoritmos más “modernos” que se tienen hoy [19].

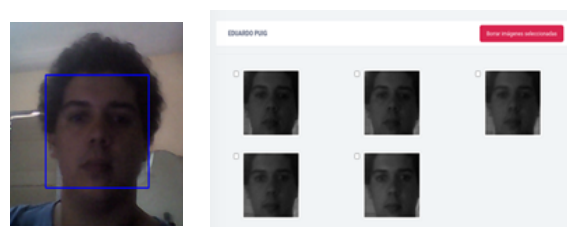


Fig. 4. Proceso de detección facial para creación del *dataset*. a) Fotograma con rostro presente encerrado en *bounding box*. b) Paso manual de revisión tras la captura automática.

Teniendo en cuenta los beneficios y desventajas, es posible usar este método de detección para la creación del dataset, puesto que se incluye también un paso de revisión manual en la aplicación web para eliminar las capturas difuminadas, los falsos positivos y, en general, cualquier fotografía que pueda empeorar el desarrollo del entrenamiento de los algoritmos subsiguientes (Fig. 4b). De este modo quedan almacenadas fotografías de los rostros de cada usuario que el sistema tratará de reconocer en una base de datos relacional de tipo PostgreSQL. Estos datos serán ingeridos luego en el flujo de trabajo big data y se le harán las transformaciones necesarias con este propósito.

B. Arquitectura big data propuesta

La acogida de las tecnologías big data ofrece una variedad de beneficios a la hora de ingerir, gestionar, analizar y extraer patrones de grandes cantidades de datos. Según [17], big data se caracteriza por tener cuatro dimensiones: volumen, variedad, velocidad y valor. El volumen hace referencia a la necesidad de procesamiento intensivo y complejo de subconjuntos de datos de gran tamaño que contienen información de valor para una organización mediante tecnologías de big data. La información de valor es cada vez más el resultado de la combinación de datos de múltiple origen y tipología que puede estar en forma estructurada, semiestructurada o no estructurada, dando al traste con la variedad de los datos.

Ahora bien, la velocidad lleva consigo la perspectiva de los datos en movimiento. La velocidad con la que se crean, procesan y analizan los datos continúa aumentando. En el contexto de big data se destacan los beneficios que se desprenden de su uso (reducción de costes, efi-

ciencia operativa y mejoras de negocio). A continuación se propone una arquitectura similar y basada en la propuesta resultante de los estudios de [18].



Fig. 5. Propuesta de arquitectura integrada de big data y FR.

En la Fig. 5 se proporciona un diagrama arquitectónico de la solución de procesamiento de big data propuesta. Se muestra un prototipo de arquitectura de extremo a extremo para ingerir, limpiar, procesar y visualizar los resultados del reconocimiento facial para generar información procesable (se consideran las imágenes un big data no estructurado, en este caso de uso). La arquitectura propuesta integra y automatiza capas de ingesta, almacenamiento, procesamiento y análisis especiales. El prototipo consta de dos flujos de trabajo principales para procesar eficazmente diferentes flujos de datos, a saber (1) canalización de big data y (2) canalización de aprendizaje automático. Se describen estos dos flujos de trabajo a continuación.

1. *Flujo de trabajo big data*: la función principal de la canalización de big data es automatizar el movimiento de datos enormes de diferentes tipos de datos desde fuentes externas de flujos de datos a un lago de datos para su análisis posterior. Además de mover datos, también se usaron bases de datos relacionales tradicionales;

así como NoSQL y almacenes de datos estáticos como archivos JPG para que estos puedan ser analizados por procesos posteriores de manera más eficiente en el *pool* de datos.

Las consideraciones claves de las herramientas y técnicas para implementar el *pipeline* de big data para la arquitectura es que puede cumplir con los siguientes objetivos: conectar múltiples tipos de datos y almacenes de archivos (por ejemplo, bases de datos relacionales y NoSQL, almacén de archivos estáticos para archivos no estructurados y no estructurados, datos semiestructurados, entre otros) y para proporcionar soporte de scripting personalizado para implementar controles de calidad de datos y preprocesamiento durante la fase de ingestión de datos.

2. *Flujo de trabajo de FR*: la función principal del pipeline de FR es ayudar a automatizar el entrenamiento y la ejecución de los algoritmos para el reconocimiento y el ajuste de parámetros de los flujos de trabajo. En otras palabras, este flujo de trabajo puede inferir el patrón o las relaciones no triviales entre un conjunto de observaciones y una respuesta deseada. Las consideraciones clave de las herramientas y técnicas para implementarlo en función de la arquitectura propuesta se enfocan en cumplir con los siguientes objetivos: apoyar la computación distribuida para que se logre una canalización escalable y proporcionar funciones e interfaces de programación de aplicaciones (APIs) para implementar características, análisis y algoritmos iterativos.

Una vez usados los dos flujos de trabajo mencionados anteriormente, la arquitectura de big data propuesta es lo suficientemente genérica como para hacer uso de diferentes fuentes de big data relacionadas con cualquier requisito de aplicación, tales como los *web crawlers*, para adquirir información de streams de datos en tiempo real y los conjuntos de datos públicos en línea.

Por su parte, Apache Spark [19] logra un alto rendimiento tanto para los datos por lotes como para los de flujo continuo, utilizando un programador de tipo Grafo Acíclico Dirigido (DAG, del inglés *Directed Acyclic Graph*) de última generación, un optimizador de consultas y un motor de ejecución físico. La investigación realizada por Hazarika et al. [20] indica que Spark puede resolver problemas de aprendizaje en grandes conjuntos de datos de imágenes utilizando bibliotecas de aprendizaje automático (ML, del inglés *machine learning*) de almacenamiento en caché iterativo en memoria que se ejecutan sobre Spark Data Frames. En algunos casos, Spark supera a Hadoop MapReduce en velocidad de cálculo diez veces en tareas de aprendizaje automático iterativas y hasta veinte veces en aplicaciones iterativas [21]. Esto se debe a que Spark aprovecha el procesamiento en memoria en comparación con MapReduce, que debe leer y escribir en el disco.

C. Sistema de reconocimiento facial

Para el análisis de los resultados obtenidos por los algoritmos que integran el sistema de FR resulta importante te-

ner en cuenta la llamada matriz de confusión (*error matrix*) [22], [23]. Esta no es más que una tabla que describe el rendimiento de un modelo supervisado en los datos de prueba, donde se desconocen los verdaderos valores. Se llama de ese modo porque a partir de su análisis es más fácil detectar dónde el sistema está confundiendo dos clases o que tipos de aciertos y errores está teniendo el modelo a la hora de pasar por el proceso de aprendizaje con los datos. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real [24].

TABLA I

Fragmento de la matriz de confusión obtenida

	Adrian	Alejandro	Anay	Ernesto	Ilgen	Jose	Julieta
Adrian	1	0	0	0	0	0	0
Alejandro	0	1	0	0	0	0	0
Anay	0	1	0	0	0	0	0
Ernesto	0	0	0	2	0	0	0
Ilgen	0	0	0	0	3	0	0
Jose	0	0	0	0	0	2	0
Julieta	0	0	0	0	0	0	1

Es posible apreciar en la Tabla 1 que la mayoría de las predicciones fueron acertadas excepto una. En ella se ha verificado que los rostros de las personas implicadas se encuentran en poses extrañas y con baja iluminación, lo que consideramos ha provocado la equivocación.

Analizando de conjunto otras métricas, como *Accuracy*, *Recall* y *F1 Score*, para el fragmento que se ha mostrado en la Tabla 1 se ha podido apreciar un desempeño destacable. En la Tabla 2 se exponen los valores obtenidos.

Una vez interpretados los resultados obtenidos se pueden apreciar que son muy similares a los estudiados durante la revisión bibliográfica que se llevó a cabo para la conformación de esta inves-

tigación. Tal es el caso de [24] que muestra cómo se desempeña un clasificador SVM, entrenado con veinte fotos de tres personas, para la detección de rostros en un video de YouTube, evidenciándose valores de precisión entre un 83 y un 95% cuando los personajes se encuentran de frente a cámara durante la corrida del video y no en posiciones de perfil que dificultan la extracción de características del rostro.

Además, está el caso citado en [24] que entrena una CNN usando *Transfer Learning* tal y como se hizo en la presente investigación, con la diferencia de que la usa para la detección de mascotas, en este caso de perros, partiendo de un conjunto de prueba del cual el algoritmo obtiene una precisión de un 80%. En [25], haciendo uso de la biblioteca de visión artificial OpenCV y de la biblioteca LIBSVM para llevar a cabo el modelo SVM, realizaron una comparación entre tres bases de datos BiID, CMU-MIT, y otra creada por el autor conformada por 200 rostros candidatos como máximo. Allí evaluaron los resultados en cuanto a la tasa de acierto y los falsos positivos, obteniéndose 97.2%, 83.1% y 93.5% de tasa de acierto, respectivamente.

Por otro lado, la tesis [26] atraviesa por las mismas tres primeras fases y algoritmos de detección facial empleados por esta investigación con la diferencia de que en la última fase para comparar y clasificar opta por la distancia euclidiana; OpenCV solicitará a la base de datos de personas los vectores almacenados (que contienen las 128 dimensiones extraídas de cada rostro). Por último, OpenCV realiza una comparación de distancias entre los vectores de rostros del video y los vectores de la base de datos de personas; de este modo se deriva el reconocimiento facial de este proceso con precisiones entre 85% y 95%, resultados que

comprenden un rango muy similar a los datos obtenidos por esta investigación.

La comparación con otros estudios para evaluar desempeño queda fuera de esta investigación ya que la base de datos fue creada para este proyecto exclusivamente. Además, su finalidad es el cotejo intrínseco de clasificadores y detectores usados, las métricas referidas y la representación de los resultados a través de la matriz de confusión.

Tabla ii

Accuracy, recall y f1 score obtenidos

	Acuary	Recall	F1 Score
Adrián	1.00	1.00	1.00
Alejandro	0.50	1.00	0.67
Anay	0.00	0.00	0.00
Ernesto	1.00	1.00	1.00
Ilien	1.00	1.00	1.00
José	1.00	1.00	1.00
Julieta	1.00	1.00	1.00
MicroAvg	0.95	0.91	0.93
Macro	0.88	0.88	0.87
Avg.			
Weighted Avg.	0.93	0.91	0.91

Como aporte esencial de la integración de las tecnologías de big data al flujo de trabajo de FR, se encuentra la mejora en los tiempos de entrenamiento respecto a la misma cantidad de datos de entrada. Haciendo uso del paralelismo en el procesamiento con la GPU y el sistema de cómputo distribuido del clúster big data de la UCLV, se ha conseguido una mejora notable: quince minutos de entrenamiento en total, usando big data, frente a 86 minutos con procesamiento convencional. Lo anterior demuestra la eficiencia de la integración de estas tecnologías y su posibilidad de uso en aplicaciones de tiempo real.

Conclusiones

En la presente investigación se propone un sistema para establecer un flujo de trabajo en función de la implementación de algoritmos de reconocimiento facial para aplicaciones en tiempo real. Para ello, se han integrado varias tecnologías con el objetivo de conseguir resultados comparables con el estado de la técnica y trabajando con un gran cúmulo de datos de entrada. La propuesta reside en el establecimiento de un sistema para creación automática de datasets orientados a FR, utilizando Django como servidor web, OpenCV y bases de datos relacionales (PostgreSQL). Los datos generados por este sistema serán procesados junto con los streams de video en tiempo real por una arquitectura big data que posibilitará un mejor desempeño del sistema de FR como el último escalón de la propuesta. Por último, se han experimentado resultados satisfactorios tras la implementación parcial del mismo y la puesta en marcha de entrenamientos y predicciones.

Referencias

- [1] A. Adouani, W. M. Ben Henia y Z. Lachiri, "Comparison of Haar-like, HOG and LBP approaches for face detection in video sequences", en *16th International Multi-Conference on Systems, Signals Devices (SSD)*, Estambul, Turquía: IEEE, mar. 2019, pp. 266-271. doi: 10.1109/SSD.2019.8893214
- [2] C. Rahmad, R. A., Asmara, D. R. H. Putra, et al., "Comparison of Viola-Jones Haar Cascade Classifier and Histogram of Oriented Gradients (HOG) for face detection", en *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 732, ago. 2020, doi: 10.1088/1757-899X/732/1/012038.

- [3] A. Ostuni. (2020, mar. 24). "Artificial Intelligence: state of the art. Where We Stand and Where We Are Going." [Internet]. Disponible en https://www.linkedin.com/pulse/artificial-intelligence-state-art-where-we-stand-going-antonio-ostuni?trk=public_profile_article_view
- [4] IBM. (s.f.). "What is Computer Vision?" [Internet]. Disponible en <https://www.ibm.com/topics/computer-vision>
- [5] Papers with Code. (s.f.). "Computer Vision." [Internet]. Disponible en <https://paperswithcode.com/area/computer-vision>
- [6] B. W. Mugalu, R. C. Wamala, J. Serugunda y A. Katumba, "Face Recognition as a Method of Authentication in a Web-Based System", *arXiv:2103.15144*, mar. 2021. [Internet]. Disponible en: <http://arxiv.org/abs/2103.15144>
- [7] J. M. Díaz, "Inteligencia artificial y Big Data como soluciones frente a la COVID-19", *Rev. Bioética y Derecho*, n.º 50, nov. 2020. [Internet]. Disponible en https://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S1886-58872020000300019
- [8] I. Jiménez Silva, *Reconocimiento facial basado en redes neuronales convolucionales*, trabajo de grado, Dpto. Teoría de la Señal y Comunicaciones, Universidad de Sevilla, España, 2018. [Internet]. Disponible en <https://idus.us.es/bitstream/handle/11441/85086/TFG-1975-JIMENEZ.pdf?sequence=1&isAllowed=y>
- [9] REC Faces. (s.f.) "Ready-Made Face Recognition Solution. [Internet]. Disponible en <https://recfaces.com/articles/como-funciona-reconocimiento-facial>
- [10] S. Z. Li y A. K. Jain, Eds., *Handbook of Face Recognition*, Nueva York: Springer, 2005.
- [11] M.-H. Yang, D. Kriegman, y N. Ahuja, "Detecting Faces in Images: A Survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, n.º 1, pp. 34-58, feb. 2002.
- [12] P. Viola y M. Jones, "Rapid object detection using a boosted cascade of simple features", en *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Kauai, HI, USA: IEEE, dic. 2001. doi: 10.1109/CVPR.2001.990517.
- [13] A. Rosebrock. (2018, feb. 26). "Face detection with OpenCV and deep learning." [Internet]. Disponible en <https://www.pyimagesearch.com/2018/02/26/face-detection-with-opencv-and-deep-learning/>
- [14] A. Rosebrock. (2019, abr. 19). "Face detection with dlib (HOG and CNN)." [Internet]. Disponible en <https://www.pyimagesearch.com/2021/04/19/face-detection-with-dlib-hog-and-cnn/>
- [15] V. Kazemi y J. Sullivan, "One Millisecond Face Alignment with an Ensemble of Regression Trees", en *Computer Vision and Pattern Recognition*, Columbus, Ohio, jun. 2014. [Internet]. Disponible en https://www.researchgate.net/publication/264419855_One_Millisecond_Face_Alignment_with_an_Ensemble_of_Regression_Trees
- [16] A. Rosebrock. (2017, abr. 3). "Facial Landmarks with Dlib, OpenCV, and Python." [Internet]. Disponible en <https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/>
- [17] J. Puyol Moreno, "Una aproximación a Big Data", *RDUNED*, n.º 14, pp. 471-506, ene. 2014, doi: 10.5944/rduuned.14.2014.13303
- [18] "Proposed Big Data Architecture for Facial Recognition Using Machine Learning", *AIMS Electronics and Elec-*

- trical Engineering*, vol. 5, n.º 1, pp. 68-92, sep. 2021, doi: 10.3934/elect-treng.2021005
- [19] Apache Spark™. (s.f.). "Unified Engine for Large-Scale Data Analytics." [Internet]. Disponible en <https://spark.apache.org/>
- [20] A. V. Hazarika, G. J. S. R. Ram y E. Jain, "Performance comparison of Hadoop and spark engine", en *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, feb. 2017, pp. 671-674. doi: 10.1109/I-SMAC.2017.8058263.
- [21] M. Zaharia, M. Chowdhury, T. Das et al., "Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing", en *NSDI'12: Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, abr. 2012, San Jose, CA, USA.
- [22] E.-J. Cheng et al., "Deep Sparse Representation Classifier for facial recognition and detection system", *Pattern Recognition Letters*, vol. 125, pp. 71-77, jul. 2019, doi: 10.1016/j.patrec.2019.03.006
- [23] S. Ajina, R. Yampolskiy, y N. Essoukri, "Evaluation of SVM Classification of Avatar Facial Recognition", en *8th International Symposium on Neural Networks*, Guilin, China, may. - jun. 2011, doi: 10.1007/978-3-642-21111-9_15
- [24] I. Siles, "Uso de AI para la detección de logos y el reconocimiento facial en ambientes de DTV", *Telemática*, vol. 20, n.º 2, ene. 2021. [En línea]. Disponible en <https://revistatelematica.cujae.edu.cu/index.php/tele/article/view/451>
- [25] D. J. Calandria, A. Cañas y A. F. Díaz, "Detección de rostros humanos mediante SVM y generación de fotografías de tipo carné", Departamento de Arquitectura y Tecnología de Computadores, Universidad de Granada, s.f.
- [26] Hamsa B. (2018, abr. 30). "Construye tu primer clasificador de Deep Learning con TensorFlow: Ejemplo de razas de perros". [Internet]. Disponible en <https://medium.com/datos-y-ciencia/construye-tu-primer-clasificador-de-deep-learning-con-tensorflow-ejemplo-de-razas-de-perros-ed218bb4df89>