ISBN: 978-628-95207-7-4. ISSN: 2414-6390. Digital Object Identifier: https://dx.doi.org/10.18687/LEIRD2023.1.1.477

DEVELOPMENT OF AN ALGORITHM OF ARTIFICIAL VISION FOR AREAS SENSITIVE IN INDUSTRIES

Ryan Abraham León León[1], Allison Nicole Bocanegra Samanamud[2], Shayna Michelle Gil Aguilar[3], and Jesús Vega Sánchez[4]

[1,4]Escuela de Ingeniería Industrial, Universidad Cesar Vallejo (UCV), Chimbote, Perú, rleonle@ucvvirtual.edu.pe; anbocanegrab@ucvvirtual.edu.pe, sgilagu@ucvvirtual.edu.pe, vvegasan@ucvvirtual.edu.pe

Abstract- In the present research work, artificial vision was applied to detect movements and faces, thus achieving greater control of sensitive areas in industries. All this through the development of algorithms, the use of the Python program, folders such as NumPy, Tkinter and a camera, since what was sought was to detect people when they are in motion, capture their faces and store them by taking photos. photos.Artificial vision allows for high efficiency since it allows working with the analysis of images of people in various sensitive areas of industries, since there are always places where access is restricted or require constant surveillance.Given this, the need to increase the efficiency and precision of control and surveillance systems arises, which works continuously for greater security.This research is intended to benefit industries in places that require constant supervision, providing an accessible solution with an efficiency of 94.60%.

Keywords: Artificial vision, algorithms, Python, facial recognition, motion detection.

Digital Object Identifier: (only for full papers, inserted by LEIRD).
ISSN, ISBN: (to be inserted by LEIRD).
DO NOT REMOVE

DEVELOPMENT OF AN ALGORITHM OF ARTIFICIAL VISION FOR AREAS SENSITIVE IN INDUSTRIES

Ryan Abraham León León[1], Allison Nicole Bocanegra Samanamud[2], Shayna Michelle Gil Aguilar[3], and Jesús Vega Sánchez[4]
[1,4]Escuela de Ingeniería Industrial, Universidad Cesar Vallejo (UCV), Chimbote, Perú, rleonle@ucvvirtual.edu.pe; anbocanegrab@ucvvirtual.edu.pe, sgilagu@ucvvirtual.edu.pe, vvegasan@ucvvirtual.edu.pe

Abstract- In the present research work, artificial vision was applied to detect movements and faces, thus achieving greater control of sensitive areas in industries. All this through the development of algorithms, the use of the Python program, folders such as NumPy, Tkinter and a camera, since what was sought was to detect people when they are in motion, capture their faces and store them by taking photos. photos.Artificial vision allows for high efficiency since it allows working with the analysis of images of people in various sensitive areas of industries, since there are always places where access is restricted or require constant surveillance.Given this, the need to increase the efficiency and precision of control and surveillance systems arises, which works continuously for greater security.This research is intended to benefit industries in places that require constant supervision, providing an accessible solution with an efficiency of 94.60%.

Keywords: Artificial vision, algorithms, Python, facial recognition, motion detection.

I. INTRODUCCIÓN

La inseguridad en el país cada vez va en aumento, no solo en las calles sino también en muchas empresas, como son los casos de robos. Todo ello se produce por la mala supervisión o vigilancia ya que hay lugares donde las empresas guardan y almacenan elementos valiosos, siendo áreas sensibles que necesitan de supervisión constante. La tecnología hoy en día está avanzando a pasos agigantados dando paso a lo que hoy se conoce como visión artificial, lo cual permite que se desarrollen programas para el área de seguridad, las cuales sean cada vez más sofisticadas y eficientes.

La visión artificial es una disciplina cuyo principal objetivo es procesar y analizar imágenes del mundo real, con la finalidad de que estas puedan ser entendidas y tratadas por un ordenador [1]. Asimismo, este campo de estudio es interdisciplinario ya que comprende conocimientos tanto de informática, matemáticas y neurociencia, capaces de interpretar y comprender información. Es por ello que este campo de estudio es interdisciplinario ya que comprende conocimientos tanto de informática, matemáticas y neurociencia, capaces de interpretar y comprender información [2]. La visión artificial está presente y revolucionando campos como los de la robótica ya la seguridad hasta la medicina y el reconocimiento facial [3]. Por ende, se trata de una tecnología que combina ordenadores con cámaras de vídeo para adquirir, analizar e interpretar imágenes de forma comparable a la inspección visual humana [4]. Empleando las redes neuronales convolucionales, se logra la clasificación de imágenes de manera sorprendente, superando la capacidad de los seres humanos [5]. La visión artificial consiste en construir un cerebro para las máquinas, con la ayuda y empleo de algoritmos [6]. El algoritmo es una abstracción matemática que describe cómo resolver un problema de manera precisa y detallada; y ejecutar diversas tareas de manera eficiente, sistemática y estructurada [7]. Asimismo, es un lenguaje de la tecnología, que permite la transformación de problemas a soluciones mediante un conjunto de reglas y procedimientos [8]. Esta herramienta es fundamental para la ciencia y tecnología permitiendo la automatización de tareas y la toma de decisiones de manera inteligente [9]. Del mismo modo, los algoritmos de visión artificial para los sistemas de seguridad permiten la detección, seguimiento e identificación de amenazas potenciales [10]. Los algoritmos empleados para realizar reconocimiento facial son muy efectivos para la identificación de individuos en entornos de seguridad detectando sospechosos o personas no autorizadas [11]. Esto mediante la segmentación de imágenes es fundamental para analizar videos e imágenes en tiempo real, logrando una detección y seguimiento de objetos en movimiento de manera precisa [12]. El reconocimiento facial es un sistema basado en el análisis facial digitalizado en 3D, que se apoya en áreas como el reconocimiento de patrones, la óptica y la visión artificial [13]. Esto, combinado con la visión artificial, se ha logrado implementar en el sistema de vigilancia inteligente capaz de detectar y alertar la presencia de sospechosos [14]. La integración de RF y VA en sistemas de seguridad ha permitido un aumento de eficiencia y precisión en la identificación de personas sospechosas ante situaciones críticas [15]. Este tema ha surgido como una tecnología prometedora con respecto a la seguridad, ya que brinda de forma eficiente y precisa la identificación biométrica en tiempo real [16]. El reconocimiento facial aplicada en la visión artificial es de gran utilidad en la detección y seguimiento de personas en entornos de seguridad [17]. El reconocimiento facial ha revolucionado la tecnología de visión artificial, logrando resultados impresionantes identificando individuos a través de imágenes y videos de vigilancia [18]. La detección de movimientos en visión artificial ha demostrado ser muy eficaz

Digital Object Identifier: (only for full papers, inserted by LEIRD).
ISSN, ISBN: (to be inserted by LEIRD).
DO NOT REMOVE

en aplicaciones de seguridad, ya que permite detectar con precisión y rapidez objetos en movimiento en escenas complejas [19]. El uso de técnicas de procesamiento informático y algoritmos de detección de objetos ha mejorado la precisión y la eficacia de las aplicaciones de seguridad [20]. Esto permite seguir y analizar objetos en movimiento para diversas aplicaciones, como la videovigilancia, la robótica y el reconocimiento de acciones humanas [21]. Este sistema está diseñado para automatizar varias tareas que ejecuta el sistema visual humano, reemplazando la supervisión humana [22]. Las redes convolucionales profundas han generado avances en el procesamiento de imágenes, video, permitiendo la detección de movimiento [23]. Proporcionando algoritmos y técnicas de extracción de características relevantes de secuencias de imágenes y el rastreo de objetos en movimientos [24]. Todo esto es posible gracias al Python, que proporciona una interfaz fácil de usar para trabajar con redes convolucionales y otros modelos de aprendizaje profundo [25]. El cual, combinado con bibliotecas como OpenCV permite que los desarrolladores implementen algoritmos de visión artificial de manera eficaz y rápida [26]. La abundancia de bibliotecas de Python centradas en la inteligencia artificial, ha permitido crear algoritmos sofisticados y construir sistemas de visión artificial precisos y eficientes [27]. Con un lenguaje de programación ideal para la inteligencia artificial con la disponibilidad de recursos e integración de tecnologías como la IoT [28]. La facilidad y versatilidad de Python lo hace ideal para crear prototipos y desarrollar algoritmos y aplicaciones de visión artificial [29]. Las bibliotecas de Python, como NumPy, SciPy y scikit-learn, proporcionan potentes herramientas para el procesamiento de imágenes y la extracción de características en tareas de visión por ordenador [30].

Con respecto a los objetivos del presente trabajo de investigación, se tiene como objetivo general desarrollar un algoritmo que permita la detección de movimiento y el rostro para áreas sensibles de la industria. Asimismo, se tiene como objetivos específicos, emplear la visión artificial en Python y librerías como NumPy y Tkinter para el desarrollo de los algoritmos.

II. Adquisición y adecuación de las imágenes

En la actualidad existen algoritmos que reducen la resolución de la imagen, ya que hay redes neuronales para el reconocimiento de objetos mediante el análisis de varias resoluciones de una misma imagen. Estas técnicas se basan en un objeto previo comparando la imagen en diferentes resoluciones, para esto se utiliza las redes neuronales las cuales se relacionan unas con otras. Estos algoritmos obtienen una imagen MxN pixels donde se han reducido sus resoluciones a 1⁄2, 1/4, etc. Para la resolución de este trabajo se calculó la media de estas e igualamos todos los resultados. Asimismo, la librería OpenCV está dirigida fundamentalmente a la visión por computador en tiempo real. Entre sus muchas áreas de aplicación destacaría: interacción hombre-máquina (HCI4); segmentación y reconocimiento de objetos; reconocimiento de gestos; seguimiento del movimiento; estructura del movimiento (SFM5); y robots móviles. Así mismo en la siguiente figura se mostrará la función y la aplicación de la carpeta opencv. (Ver Fig. 1)

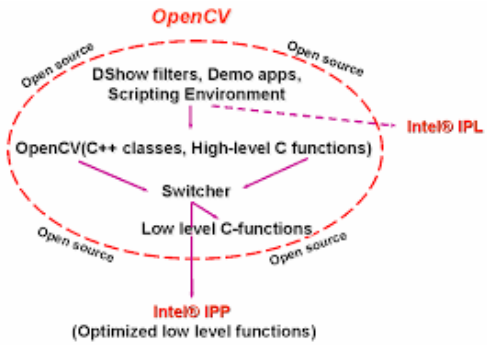

Figura 1: Diagrama de Open CV en esta imagen nos muestra el funcionamiento de la carpeta dentro del proyecto.

La librería OpenCV proporciona numerosos elementos de alto nivel, como veremos en secciones posteriores, que facilitan sobremanera el trabajo al usuario como, por ejemplo: (ver Fig. 2)

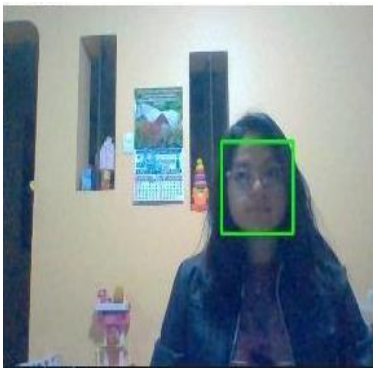

Figura 2: Reconocimiento facial.

También los entornos de scripting hacen uso de estas funciones para implementar su funcionalidad. La librería OpenCV proporciona una gran diversidad de entornos. En la sección 2.2 detallamos de forma pormenorizada algunos de los más utilizados.

Todas estas herramientas de alto nivel hacen uso de un paquete de clases C++ y funciones C de alto nivel que utilizan a su vez funciones muy eficientes escritas en C. Concretamente, el conjunto de funciones suministradas por la librería OpenCV se agrupan en los siguientes bloques, la cual muestra a detalle el uso e importancia de esta librería para poder realizar la codificación del proyecto en Python. (Ver Fig. 3)

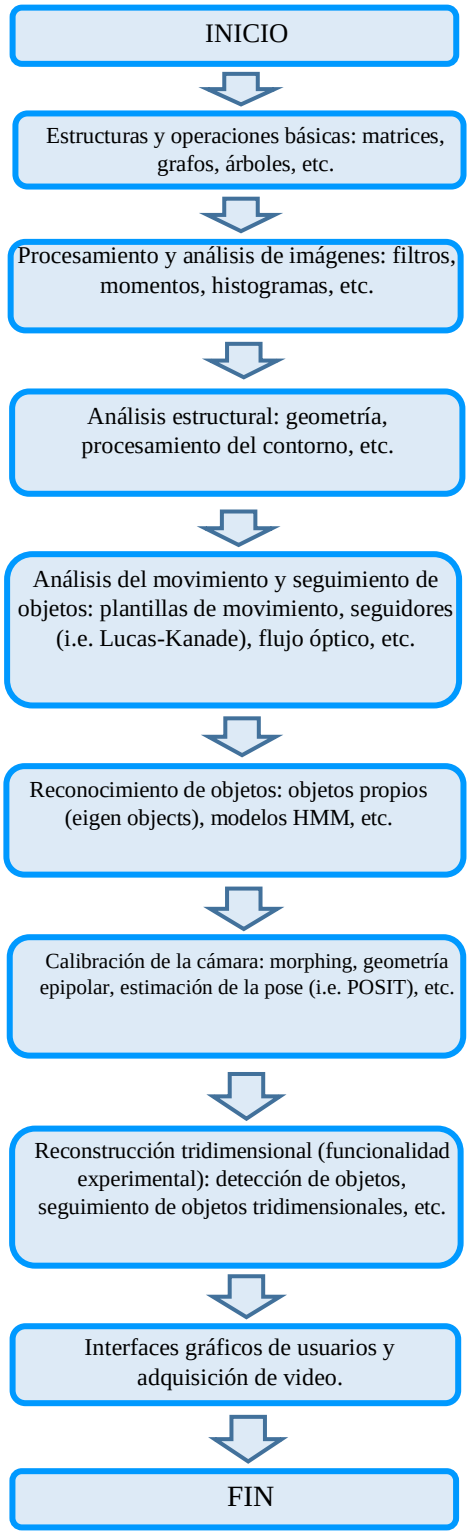

Figura 3: Diagrama de Flujo del conjunto de funciones por la librería Opencv.

La siguiente imagen muestra a detalle como identifica mediante un cuadro rojo el movimiento y rostro de la persona. (Ver Figura 4.)

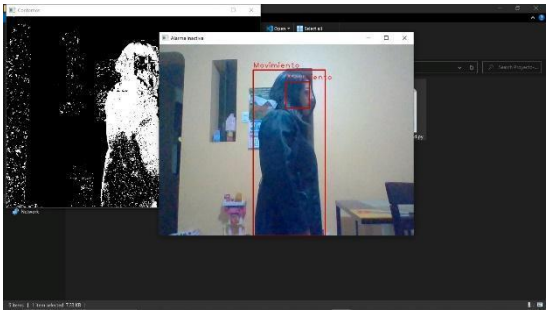

Figura 4: Captura de detección facial.

Una de las primeras extensiones que se programaron para Python, quizás por ser una de las más necesarias, fue la que proporcionaba la clase de array n-dimensional. Un array es un tipo que puede describirse mediante dos características: Sus dimensiones, el tipo de sus elementos; esto implica que un array, a diferencia de las listas o los tuples, es homogéneo en memoria. Un array es gracias al módulo Numpy otra clase de Python y la siguiente codificación fue aplicada al proyecto realizado. (Ver Fig. 5)

```
#Aplicar lector
mascara = mov.apply(Frame)

#Copia para detectar entornos
contornos = mascara.copy()

#Buscamos los contornos
con, jerarquia = cv2.findContours(contornos, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

#Pasamos por los contornos
for C in con:
    #Elimiinamos ruido(Contornos pequeños)
    if cv2.contourArea(c) <1500:
        #com.write(i.encode('ascii'))
        continue
    if cv2.contourArea(c) >= 1500:
        #Obtenemos los límites
        (x, y, w, h) = cv2.boundingRect(c)
        #Dibujamos el rectangulo
        cv2.rectangle(Frame, (x, y), (x+w, y+h), (0, 0, 225), 2)
        cv2.putText(Frame, '()', format('Movimiento'), (x, y-5), 1, 1.3, (0, 0, 255), 1, cv2.LINE_AA)
```

Figura 5: Empleo del programa NUMPY

El algoritmo de la investigación se compone de tres etapas cruciales. En la primera fase, se realiza la instalación de las bibliotecas necesarias (numpy, cv2), las cuales son conjuntos de módulos y funciones prediseñadas que brindan a los desarrolladores acceso a diversas funcionalidades sin requerir la creación de código desde cero. La segunda etapa se enfoca en la detección de movimientos utilizando OpenCV. En este punto, empleamos el algoritmo createBackgroundSubtractorKNN de OpenCV para identificar movimientos en una secuencia de video. Finalmente, el último paso involucra la detección y extracción de contornos. En esta fase, aprovechamos la función findContours de OpenCV para detectar los contornos presentes en la máscara de primer plano y mediante esta metodología en tres fases, se logra un enfoque integral que aborda desde la preparación inicial de las herramientas hasta la detección precisa de movimientos y contornos en el flujo de video en cuestión. (Ver fig. 6)

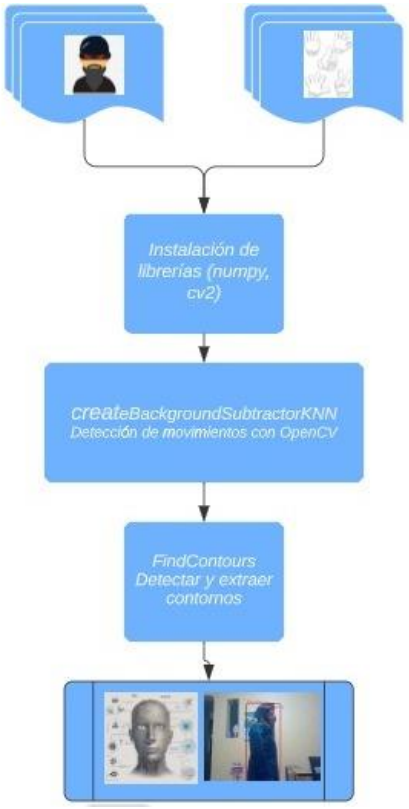

Figura 6: Tres etapas cruciales del algoritmo.

III. RESULTADOS

El desarrollo de soluciones de visión artificial para la detección de movimiento y reconocimiento facial se destaca por el uso principal de Python como lenguaje de programación, gracias a su amplia gama de herramientas especializadas. La implementación implica una codificación minuciosa con algoritmos avanzados y abarca desde la manipulación de datos de imágenes hasta la optimización de algoritmos para lograr resultados precisos. En resumen, el proceso combina habilidades técnicas, creatividad y conocimiento de Python para sistemas eficaces de visión artificial. Se presenta el empleo de Python y la codificación. (Ver Fig. 7)

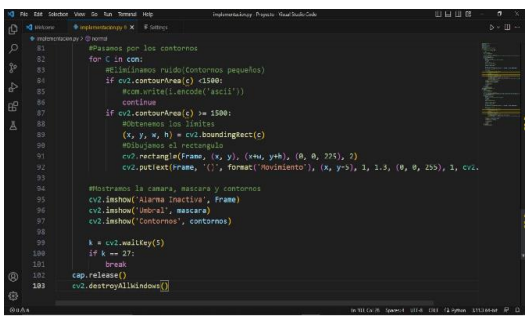

Figura 7: Empleo de Python y la codificación

Al digitar los códigos en los programas anteriormente mencionados se logró la detección de movimiento como se muestra. Así mismo se observa cómo aparece un cuadro verde identificando rostro y tomando la captura según el programa. (Ver Fig.8).

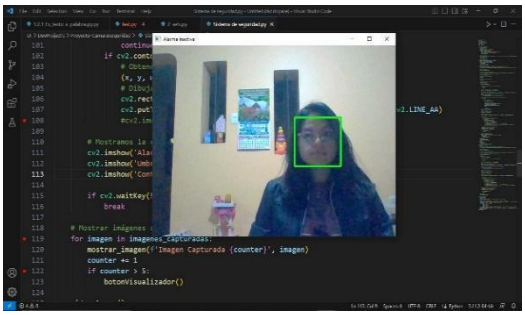

Figura 8: Funcionamiento del programa al detectar movimiento.

Posteriormente, mediante la incorporación de códigos adicionales en los programas, se alcanzó con éxito la capacidad de realizar la detección de rostros y el reconocimiento facial de manera eficiente y precisa. Esta fase de desarrollo y refinamiento de algoritmos permitió que el sistema identificara y caracterizara con precisión los rostros en el conjunto de datos, lo que representa un logro significativo en la aplicación de la visión artificial a esta tarea específica (Ver Fig. 9).

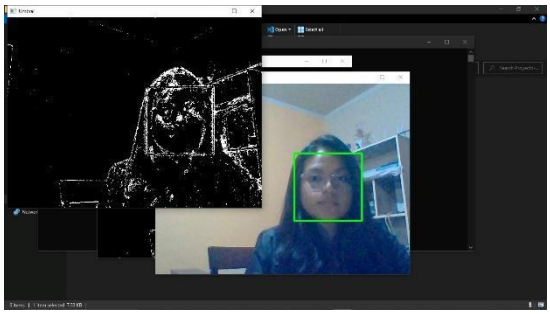

Figura 9: Funcionamiento del programa al detectar rostros

En el proceso de llevar a cabo estas tareas, se ha realizado una meticulosa configuración del programa. Este algoritmo está diseñado para detectar tanto movimientos como rostros y, al hacerlo, captura imágenes a una velocidad de 5 por segundo. Estas imágenes se almacenan de manera ordenada en una carpeta en el dispositivo utilizado, que en este caso es una laptop. Además, se ha llevado a cabo un total de 50 pruebas exhaustivas que han arrojado resultados con una tasa de precisión del 94.60%. Este logro no solo refleja un buen desarrollo del sistema en Python, sino que también abre nuevas puertas para futuras aplicaciones en campos como la seguridad y la vigilancia. Destaca el potencial que la visión artificial tiene para abordar desafíos prácticos y resolver problemas en diversas industrias, estableciendo un hito en la investigación y desarrollo tecnológico.

Tenemos la tabla 1: N° de pruebas que reflejan la precisión de 94.60 % en 50 pruebas.

TABLA 1
Números de pruebas de la Precisión

N° DE MUESTRA	NO FALLARON	FALLARON	PRECISIÓN	EXPLICACIÓN DE FALLAS
10	5	5	70.38%	Fallas de detención en movimiento
20	14	6	77.89%	Fallas de movimientos faciales
30	25	5	85.65%	Fallas de capturas
40	36	4	88.54%	Fallas en movimiento
50	45	5	94.60%	Fallas en tiempo de captura

IV. Discusión

En la investigación visión artificial aplicada a la detección e identificación de personas en tiempo real [31] en donde se utilizó una red neuronal Convolucional como estrategia que resuelve de manera óptima los objetivos establecidos para el proyecto, teniendo un 90% de precisión y exactitud. En nuestra investigación solo hemos usado técnicas de visión artificial para el procesamiento de imágenes, pero el resultado fue de manera efectiva. Al aplicarlo en esta investigación en donde el número de muestra fue de 50 el cual nos da 94.60% de precisión gracias a la función que Python brinda como la biblioteca OPENCV, la cual es una librería que está dirigida fundamentalmente a la visión por computador en tiempo real. Aunque las redes neuronales convolucionales son potentes y versátiles para muchos problemas de visión, hay situaciones específicas en las que las técnicas de visión artificial pueden ser preferibles debido a su eficiencia, requisitos de datos, interpretabilidad u otros factores contextuales.

V. Conclusiones

En la presente investigación se logró desarrollar el algoritmo necesario para la detección de movimiento y rostro para áreas sensibles de la industria. También logró emplear la visión artificial para la mejora de la seguridad en áreas sensibles de la industria, permitiendo guardar un registro de imágenes mediante la toma de fotos.

Se empleó el programa Python y las carpetas como NumPy y Tkinter para el desarrollo de los algoritmos y su funcionamiento con éxito, detectando los movimientos y rostros de las personas que eran captadas por la cámara.

Se concluye que para el área de seguridad es necesario un resguardo y vigilancia constante, no solo basta con colocar personal en cada turno ya que se requiere de protección y monitoreo. Es por ello que se debe optar por la inteligencia artificial y el desarrollo del algoritmo planteado en esta investigación, pues permiten una mayor eficiencia en la seguridad ya sea en empresas u hogares.

Referencias

[1] B. Borrella (2022, Septiembre) Introducción a la visión artificial: procesos y aplicaciones. [En línea]. Recuperado de: https://docta.ucm.es/entities/publication/072ca3fb-540f-4dc9-8a16-0241cb5cd986

[2] R. Szeliski, *Computer vision: algotithms and applications.* Springer Science & Business Media, 2ª ed.The University of Washington,2022. . [En línea]. Recuperado de: http://cv2.csie.ntu.edu.tw/CV2/2023/textbook.pdf

[3] D. Forsyth y J. Ponce, *Computer vision: a modern approach. Pearson* . 2ª ed,2008. [En línea]. Recuperado de: https://dokumen.tips/documents/computer-vision-a-modern-approach-2nd-ed-d-forsyth-j-ponce-pearson-2012-bbs.html?page=1

[4] M. Cajas y P. Viri, (2017, Marzo). Diseño e implementación de un sistema de seguridad vehicular mediante reconocimiento facial a través de visión artificial. [En línea]. Recuperado de: https://dspace.ups.edu.ec/bitstream/123456789/13566/1/UPS-CT006920.pdf

[5] A. Krizhevsky, I. Sutskever y G. Hinton. (2012). ImageNet classification with deep convolutional neural entworks. [En línea]. Recuperado de: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[6] A. Torralba. (2022, Diciembre 30). Visión artificial. [En línea]. Recuperado de: https://www.farodevigo.es/sociedad/2016/12/04/antonio-torralba-robots-supera ran-vision-16446763.html

[7] S. Skiena. (2008, Enero 15). The Algorithm Desing Manual. [En línea]. Recuperado de: https://mimoza.marmara.edu.tr/~msakalli/cse706_12/SkienaTheAlgorithmDesi gnManual.pdf

[8] A. Levitin. (2012). Introduction to the Desing and Analysis of Algorithms. Pearson. 3ª ed. [En línea]. Recuperado de: https://doc.lagout.org/science/0_Computer%20Science/2_Algorithms/Introduction%20to%20the%20Design%20and%20Analysis%20of%20Algorithms%20%283rd%20ed.%29%20%5BLevitin%202011-10-09%5D.pdf

[9] D. Dasgupta, C. Papadimitriou. y U. Vazirani. (2006, Julio 18). Algorithms. McGraw-Hill Education. [En línea]. Recuperado de: http://algorithmics.lsi.upc.edu/docs/Dasgupta-Papadimitriou-Vazirani.pdf

[10] J. Liu, X. Li, J. Zhang y M. Zhou. (2015 Septiembre 11). Deep Learning for Visual Understanding: A Review. Neurocomputing. [En línea]. Recuperado de: https://deeplearning.ir/wp-content/uploads/2016/07/Deep-learning-for-visual-understanding_-A-review.pdf

[11] B. Chacua. (2019, Diciembre 14). Diseño de un Sistema Prototipo De Reconocimiento facial para la identificación de personas en la facultad de Ingeniería en Ciencias Aplicadas (FICA) de la universidad Técnica del Norte utilizando técnicas de inteligencia artificial. [En línea]. Recuperado de: http://repositorio.utn.edu.ec/handle/123456789/9572

[12] H. Perez. (2011, Mayo 24). Detección y Seguimiento de Personas Basado en Vectores de Movimiento. [En línea]. Recuperado de:https://www.repositoriodigital.ipn.mx/handle/123456789/8051

[13] M. Vásquez. (2014, Mayo). Sistema de reconocimiento facial mediante técnicas de visión tridimensional. [En línea]. Recuperado de :http://cio.repositorioinstitucional.mx/jspui/handle/1002/436

[14] Z. Zhang. (2014, Septiembre). Detección de puntos de referencia faciales mediante aprendizaje multitarea profundo. [En línea]. Recuperado de:https://www.researchgate.net/publication/264786906_Facial_Landmark_Detection_by_Deep_Multi-task_Learning

[15] A. Roca. (2021, Noviembre 18). Sistema de detección de personas y cálculo de distancias euclidianas mediante el uso de redes neuronales convolucionales integrando OpenCV y CUDA para medir el índice de cumplimiento del distanciamiento social de los estudiantes de la EEB Lcda. Angélica Villón Lindao. [En línea]. Recuperado dee:https://repositorio.upse.edu.ec/handle/46000/6484

[16] L. Fierro. (2011, mayo)Prototipo de un sistema de seguridad para la estancia infantil del IIT integrando tecnología RFID y cámaras IP. [En línea]. Recuperado de: http://erecursos.uacj.mx/handle/20.500.11961/3076

[17] Z. Stan. (2011, Agosto 22) Manual de Reconocimiento Facial. Springer Londres (2nd ed.). [En línea]. Recuperado de:https://link.springer.com/book/10.1007/978-0-85729-932-1

[18] F. Morcillo. (2020, Diciembre 10). Desarrollo de un sistema de reconocimiento facial utilizando Deep Learning con OpenCV. [En línea]. Recuperado de: https://riunet.upv.es/handle/10251/156694

[19] J. Redmon. (2018, Abril 8). Una mejora Incremental. [En línea]. Recuperado de: https://arxiv.org/abs/1804.02767

[20] J. Córdova. (2021, Agosto 12). Aplicación de Inteligencia Artificial para monitorear el uso de mascarillas de protección. [En línea]. Recuperado de: http://www.scielo.org.co/scielo.php?pid=S1900-65862021000100205&script=sci_arttext

[21] A. Yilma. (2006, Diciembre). Object tracking: a survey. ACM Comput Surv. [En línea]. Recuperado de: https://www.researchgate.net/publication/220566062_Object_tracking_a_survey_ACM_Comput_Surv

[22] R. Devashree. (2022, Marzo 29). Detección de anomalías utilizando edge computing en sistemas de videovigilancia: revisión. [En línea]. Recuperado de: https://link.springer.com/article/10.1007/s13735-022-00227-8

[23] Y. LeCun. (2015, Mayo 27). Aprendizaje profundo**.** [En línea]. Recuperado de: https://www.nature.com/articles/nature14539

[24] R. Szeliski. (2022, Enero 3). Visión artificial : algoritmos y aplicaciones, doi: https://doi.org/10.1007/978-3-030-34372-9

[25] F. Chollet. (2017, Julio 17). Deep Learning with Python. [En línea]. Recuperado de: https://deeplearning.lipingyang.org/wp-content/uploads/2016/12/The-limitations-of-deep-learning_part1.pdf

[26] G. Bradski. (2008, Julio 12). Learning OpenCV: Computer Vision with the OpenCV Library. [En línea]. Recuperado de: https://www.bogotobogo.com/cplusplus/files/OReilly%20Learning%20OpenCV.pdf

[27] A. Géron. (2019 Septiembre). Aprendizaje automático práctico con Scikit-Learn, Keras y Tensor Flow. [En línea]. Recuperado de: https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/

[28] R. Rosebrock. (2017, Agosto). Aprendizaje profundo para visión artificial con Python: paquete de inicio. [En línea]. Recuperado de: https://dokumen.pub/deep-learning-for-computer-vision-with-python-starter-bundle.html

[29]J. Solem. (2012, Marzo 18). Programando visión artificial con Python. [En línea]. Recuperado de: http://programmingcomputervision.com/

[30] A. Pérez. (2018, Febrero 01). Sistema de visión industrial para piezas en procesos con dispositivo móvil o single board computer: Revisión de literatura. [En línea]. Recuperado de: https://jakevdp.github.io/PythonDataScienceHandbook/

[31] E. ACUÑA y A. Diego. Visión artificial aplicada a la detección e identificación de personas en tiempo real. 2019. Tesis de Licenciatura. Quito, 2019. Recuperado de: http://bibdigital.epn.edu.ec/handle/15000/20098