

SISTEMA DE RECONOCIMIENTO FACIAL Y DE CUBREBOCAS APLICANDO HERRAMIENTAS MATEMÁTICAS PARA DETECTAR EL USO DE MASCARILLAS DE BIOSEGURIDAD COMO MEDIDA DE PREVENCIÓN DE COVID-19

Ronald Paúl Andrade Pérez¹, Omar Gustavo Bravo Quezada²

¹ Universidad Politécnica Salesiana Cuenca

E-mail: randradep1@est.ups.edu.ec  <https://orcid.org/0000-0003-0443-9451>

² Universidad Politécnica Salesiana Cuenca

E-mail: obravo@ups.edu.ec  <https://orcid.org/0000-0002-5543-5672>

Fecha de recepción:
18/09/2020

Fecha de aceptación:
20/01/2021

RESUMEN

El presente trabajo está inspirado en la necesidad de precautelar la propagación del coronavirus, o también denominado COVID-19. Se presenta una propuesta de un prototipo de reconocimiento de rostros con un detalle adicional, la de identificar a personas que no están utilizando cubrebocas, elemento de bioseguridad tan necesario para evitar la propagación de esta enfermedad, sistema que ha sido desarrollado bajo la plataforma del software matemático avanzado llamado Matlab. Esta iniciativa nace de la necesidad de realizar un proyecto integrador dentro de la rama de las matemáticas avanzadas en la que era necesario utilizar este software, aplicando una herramienta interna de la transformada de Fourier, aplicación que encaja como una ayuda tecnológica importante dentro de un contexto de pandemia mundial. Se ha creído conveniente realizar este reconocimiento para detectar personas que no están utilizando la mascarilla de bioseguridad

en cualquier espacio público, evitando de esta forma la posible propagación del coronavirus. Nuestra aplicación captura imágenes, realiza transformaciones de Fourier, transformada Z entre otras, las cuales son modificadas a una escala de grises y ejecuta comparaciones con datos óseos y frecuencias previamente verificadas y estudiadas, para al final se comparan con una base de datos previamente obtenida, dando como resultado por un lado la identificación de personas y, por otro lado, la verificación si una persona está o no utilizando una mascarilla de bioseguridad, con márgenes de error muy bajos y que pueden ser comprobables. Los resultados de nuestros experimentos dan a conocer que esta propuesta puede ser factible de implementarla para un sinnúmero de proyectos en contextos de pandemia.

Palabras Clave: covid-19, cubrebocas, reconocimiento facial, transformada de fourier, frecuencias de rostros, escala de gris, matlab.

ABSTRACT

This work is inspired by the need to protect the spread of the coronavirus, or also

called COVID-19. A proposal for a face recognition prototype is presented with an additional detail, that of identifying people who are not using face masks, a biosafety element so nec-

essary to prevent the spread of this disease, a system that has been developed under the software platform advanced mathematician called Matlab. This initiative arises from the need to carry out an integrative project within the branch of advanced mathematics in which it was necessary to use this software, applying an internal Fourier transform tool, an application that fits as an important technological aid within a context global pandemic. It has been considered convenient to carry out this examination to detect people who are not using the biosafety mask in any public space, thus avoiding the possible spread of the coronavirus. Our application captures images, performs Fourier transforms, Z transforms among others, which

are modified to a gray scale and performs comparisons with previously verified and studied bone data and frequencies, the same ones that are ultimately compared with a database previously obtained, resulting on the one hand the identification of people and on the other hand, the verification whether or not a person is using a biosafety mask, with very low margins of error and that can be verifiable. The results of our experiments reveal that this proposal may be feasible to implement for countless projects in pandemic contexts.

Keywords: *covid-19, face mask, facial recognition, fourier transform, face frequencies, gray scale, matlab.*

Forma sugerida de citar: Andrade Pérez, R. & Bravo Quezada, O. (2021). Sistema de reconocimiento facial y de cubrebocas aplicando herramientas matemáticas para detectar el uso de mascarillas de bioseguridad como medida de prevención de COVID-19. Convergence Tech Revista Científica. 5(1), 53-60. <https://doi.org/10.53592/convtech.v5iV.1>

INTRODUCCIÓN

La enfermedad del coronavirus COVID 19 se fue detectada por primera vez en la ciudad China de Wuhan a finales del año 2019 (Gallegos, 2020), la misma que se expandió rápidamente por todo el mundo, llegando a territorio ecuatoriano durante el mes de febrero de 2020, situación que infirió para que las autoridades gubernamentales ordenen una cuarentena nacional a mediados del mes de marzo del mismo año para evitar una posible expansión exponencial de esta enfermedad (Haro, 2020). Por otro lado, la Organización Mundial de la Salud (OMS) determinó que las medidas de bioseguridad más relevantes para la no propagación de este virus se resumían en tres aspectos fundamentales: que toda persona debía cumplir con el distanciamiento social, lavado frecuente de manos y el uso obligatorio de cubrebocas o comúnmente denominada "mascarilla". En este contexto y bajo la necesidad de realizar un proyecto integrador dentro de la cátedra de Matemáticas Avanzadas, específicamente aplicando la transformada de Fourier, se ha presentado una propuesta de proyecto para el reconocimiento facial y detección de personas que no lleven consigo un cubrebocas, elemento tan necesario para evitar el contagio masivo de este virus en la población.

Bajo esa premisa, podemos indicar que el reconocimiento facial se viene desarrollando e investigando desde hace algunos años atrás y es un área de investigación que tiene un sinnúmero de aplicaciones en la cotidianeidad. Podemos definirla como "una forma de autenticación biométrica que utiliza medidas corporales para verificar una identidad mediante la medición de la forma y estructura únicas de rostros" (Revelo Álava, 2019). Básicamente es necesario la utilización de cámaras en la que en la mayoría de los sistemas de reconocimiento facial dependen de imágenes en dos dimensiones (2D), debido a que generalmente las cámaras no tienen la capacidad de capturar información de profundidad como la longitud de la nariz o la profundidad de la cavidad ocular, o simplemente porque la base de datos de referencia consiste en imágenes en 2D, como fotos policiales o de pasaporte (Díaz, 2004). El reconocimiento facial 2D utiliza principalmente puntos de referencia como la nariz, la boca y los ojos para identificar un rostro, y mide tanto el ancho y la forma de los rasgos, como la distancia que existe entre ellos en la cara (Pérez, 2012). Estas mediciones son convertidas luego en un código numérico los mismos que son resueltos con cálculos matemáticos aplicando transformadas: la transformada de Fourier y la transformada Z.

Cabe destacar que, dentro de nuestra propuesta también fue importante determinar el margen de error máximo y mínimo existente durante el reconocimiento facial, para así compararlos con los resultados calculados y de esta forma se los pueda cotejar determinando los puntos en donde se debe empezar a recorrer las imágenes captadas, obteniendo los puntos más importantes de los ojos, nariz y la boca.

Nuestra propuesta puede ser implementada para controlar el uso de mascarillas en distintos accesos de diferentes puntos dentro de una ciudad, tales como: accesos a Hospitales, oficinas públicas y/o privadas, ingreso a transporte público, principalmente ingreso al Tranvía de la Ciudad de Cuenca, etc., evitando la presencia de personas para controlar esta situación, las cuales pueden estar realizando otras actividades, siendo este un sistema autónomo, previniendo contagios y evitando una masiva propagación.

DESARROLLO

Para el desarrollo de nuestro proyecto, hemos utilizado el software matemático denominado Matlab en su versión r2016b, el mismo que contiene herramientas ya programadas internamente y que han sido utilizadas para la elaboración de nuestra propuesta tecnológica.

Estas herramientas, a las que también se las conoce como métodos, se las detalla a continuación y con sus respectivas características:

- **Size():** devuelve los tamaños de cada dimensión de la matriz x en un vector d con $n = \text{dims}(x)$ elementos. Si x es un escalar, que Matlab considera como una matriz 1 por 1, $\text{size}(x)$ devuelve el vector $[1 \ 1]$.
- **Algoritmo Viola-jones**
“El algoritmo de Viola-Jones nace como resultado de un entorno de trabajo (framework) de Viola-Jones para reconocimiento de objetos, siendo este la primera solución capaz de devolver una tasa de detecciones aceptables en tiempo real” (Mur Igualada, 2015) y fue propuesto por Paúl Viola y Michael Jones en 2001. Este framework se puede entrenar para reconocer gran variedad de objetos, pero la principal motivación de su creación fue la detección facial. Sus principales características son una tasa de detección muy alta, con una tasa de falsos positivos muy baja, es capaz de trabajar en tiempo real (mínimo 2 frames por segundo), y únicamente es indicado para la detección facial, no el reconocimiento (es capaz de encontrar caras, pero no de distinguirlas) (Caballero Barriga, 2017). El algoritmo consta de 4 etapas que a continuación las detallamos:
 - La primera etapa consiste en la obtención de descriptores Haar-like, en la que la detección facial se basa en las similitudes que comparten todas las caras, y que quedan reflejadas en imágenes con única diferencia en la intensidad que poseen los píxeles, con los que se pueden construir filtros rectangulares basados en estas diferencias.
 - En la segunda etapa, mediante un proceso de representación de imágenes denominado Integral Image, se evalúan estos descriptores de una forma considerablemente rápida frente a otras alternativas (Valvert Gamboa J, R., 2016).
 - En la tercera etapa se emplea el algoritmo de aprendizaje AdaBoost para seleccionar las mejores características en un conjunto de imágenes y entrenar un clasificador, ya que de otra forma resultaría muy costoso (en una imagen de 24×24 píxeles hay un total de 162.336 posibles descriptores) (Mur Igualada, G., 2015). De esta forma se construye un clasificador duro como una combinación lineal de clasificadores blandos con pesos aplicados.
- **Imread:** Lee la imagen del archivo especificado por filename, deduciendo el formato del archivo a partir de su contenido, si filename es un archivo de varias imágenes.
- **Vision.CascadeObjectDetector():** el detector de objetos en cascada utiliza el algoritmo viola-jones para detectar rostros, narices, ojos, boca o parte superior del cuerpo de las personas.
- **Step:** calcula la respuesta escalonada de un sistema dinámico. Para el caso del espacio de estado, se supone un estado inicial cero. Cuando se invoca sin argumentos de salida, esta función traza la respuesta del paso en la pantalla.

$$h(x) = \text{sign}(\sum_{j=1}^n a_j h_j(x)) \quad 1$$

Cada uno de estos clasificadores blandos es una función umbral basada en un descriptor.

$$h_j(x) = \text{sign}(\{-s_j \text{hiff}_j < 0\}) \quad 2$$

El valor del umbral, así como la polaridad y los coeficientes se determinan durante el entrenamiento del clasificador.

- La cuarta etapa constituye la arquitectura en cascada, la misma que consiste en definir el nombre de la clase empleada en Matlab Vision. CascadeObjectDetector. Con esta arquitectura se obtienen unos resultados bastante aceptables, aun teniendo unos resultados pobres en cada clasificador de forma individual: Una estructura de 32 niveles puede obtener una tasa de falso positivo de 10⁻⁶ con una tasa de clasificador individual de 65% (Jin, Z., Lou, Z., Yang, J., Sun, Q. 2007). En el otro lado nos encontramos que para obtener una tasa de detección de aproximadamente 90%, cada clasificador individual debe tener una tasa de detección de 99.7%.

Así pues, el algoritmo de Viola-Jones presenta las siguientes ventajas:

- Gran velocidad de procesamiento.
- Selección de descriptores eficiente.
- No depende de localización ni escala.
- Trabaja con descriptores, en lugar de la imagen.

Transformada de Fourier Para Detección de Rostros

La transformada de Fourier está dada por la siguiente ecuación:

$$F[u] = \int f[t] e^{-i2\pi ut} dt \quad 3$$

La Ecuación 3 corresponde a la transformada de Fourier de una señal continua $j(t)$, la cual presenta algunas deficiencias debido a que esta transformación no está

localizada en el dominio espacial. Debido a ello no puede representar adecuadamente los cambios que suceden en la señal. Para solucionar este problema, Gabor adaptó la transformada de Fourier para analizar solo una pequeña porción de la señal en un tiempo, proceso llamado transformada por ventanas de Fourier (WFT) (Daubechies I, 1990). En este caso una ventana de observación se desplaza en el dominio de tiempo, y se calcula la transformada de Fourier a partir de la porción visible de la señal para cada posición de la ventana. Considerando $j(t)$ como la función que cumple el papel de ventana, se puede definir la transformada por ventanas de Fourier de una señal continua $f(t)$ como:

$$F[u, b] = \int j[t-b] f[t] e^{-i2\pi ut} dt \quad 4$$

Base de Datos Oracle

Oracle constituye una herramienta de gestión de bases de datos muy popular, que ha sido utilizada de forma global por un sinnúmero de empresas alrededor del mundo para poder controlar y gestionar grandes cantidades de información desde un solo archivo. Las ventajas de poder usar esta base de datos se dan por la utilización de tecnologías muy avanzada en las que se puede controlar o vigilar quienes son los que accede a estos datos, siendo esta una herramienta de fácil uso ya que es muy intuitiva.

Enfocándonos en la seguridad de esta herramienta, podemos mencionar que protege toda la información de una manera muy eficaz, entregándonos una alta seguridad y fiabilidad inclusive al realizar copias de seguridad (Oracle, 2020). Adicionalmente, esta no permite que la información sea redundante, aunque también depende de cómo se configure sus atributos, de cómo este programada sus tablas y en donde se desee guardar los datos.

Otra ventaja que podemos mencionar es que constituye una base de datos relacional, lo que significa el almacenar y proporcionar accesos a puntos de datos que están relacionados entre sí. Las bases de datos relacionales tienen como base el modelo relacional, que significa que tiene una forma intuitiva y directa de representar datos en las tablas, en la que en cada fila de la tabla constituye un registro con un identificador ID único, llamado clave. Mientras tanto, todas

las columnas de la tabla contienen los llamados atributos de los datos, y cada registro tiene un valor para cada atributo, lo que hace fácil la creación de las relaciones entre los distintos puntos de datos.

Por otro lado, podemos mencionar que Oracle nos da la posibilidad de tener una gran consistencia de todos los datos por las características ya mencionadas anteriormente, aunque en ciertas ocasiones pueden surgir algunos conflictos dentro de una base de datos debido a que varios usuarios o aplicaciones intentan cambiar los datos durante un mismo instante de tiempo pudiendo ser modificada su estructura original, por lo que es de mucha importancia utilizar técnicas de bloqueo cuya acción es la de minimizar la posibilidad de conflictos y de esta manera poder mantener la total integridad de los datos guardados.

Estas técnicas de bloqueo lo que hacen es evitar que tanto usuarios como aplicaciones, pueda acceder a los datos mientras se están actualizando. Algunas bases de datos, este bloqueo se aplica a todo el conjunto de la tabla, creando un rendimiento totalmente negativo de la aplicación, mientras que, en otras bases de datos, como relacionales de Oracle, lo que hacen es aplicar bloqueos a nivel de sus registros, dejando disponibles los otros registros dentro de la tabla, lo que permite garantizar un alto rendimiento de la aplicación (Oracle, 2020).

Finalmente, hemos decidido utilizar esta base de datos para el registro de los usuarios que van a ser detectados con la finalidad de que nuestra aplicación pueda acceder de manera fácil y eficiente para reconocer rostros y personas que no tengan consigo el cubrebocas.

Oracle en el Sistema de Reconocimiento Facial

Una vez que se ha explicado algunas de las razones del porqué estamos utilizando Oracle, detallaremos la forma que hemos conseguido conectarnos entre el programa desarrollado y esta base de datos la misma que contendrá el almacenamiento de toda la información requerida y así, de esta manera, poder realizar las comparaciones necesarias con la finalidad de que se nos sea devuelto los resultados deseados de reconocimiento.

Entonces, para conectarnos a la base de datos Oracle mediante Matlab, usaremos la siguiente línea de código en la que se especificará el host, el puerto predeterminado de

Oracle para todas aquellas conexiones que se desean acceder. El puerto en mención es el 1521 que prácticamente es el mismo para todos los lenguajes de programación, en el que debe ser requerido tanto el nombre de usuario como la contraseña, tal como se muestra a continuación:

```
conn=database('XE','usuario','contrasenia','Vendor','Oracle','DriverType','thin','Server','localhost','Port-Number',1521).
```

Para cerrar la conexión establecida, lo cual es muy importante para que no se produzcan errores, es el siguiente código:

```
close(conn);
```

Este código es el utilizado en una conexión entre el software Matlab con la base de datos Oracle.

Recordemos que nuestra propuesta consiste en dos aplicaciones. La primera la de reconocer un rostro de persona con el objetivo de identificarla y, la segunda, la de reconocer si una persona está o no con el cubrebocas, con la finalidad de llamarle la atención para que sea consciente de su uso obligatorio, o simplemente no permitirle el ingreso a un determinado lugar.

En este contexto, si nuestra propuesta fuese aplicada en un entorno de seguridad, debería existir previamente una base de datos con los nombres y apellidos, por ejemplo, de personas peligrosas o que estén siendo buscadas por las autoridades, a quienes se les intenta reconocer. Para nuestro prototipo, es necesario ingresar los datos de las personas con las que vamos a realizar nuestros experimentos.

En base a lo descrito anteriormente debemos mencionar que, durante la ejecución del programa en la base de datos que se está creando van a existir dos tablas, siendo estas:

- **Persona:** la misma que guarda los datos del usuario al cual se le ha capturado el rostro, los datos que guarda son los nombres y apellidos.
- **Imágenes:** se guardará los datos de la ubicación de la carpeta en donde se ubicaran las imágenes, la frecuencia del rostro y la imagen.

La frecuencia del rostro significa que se aplicará las técnicas de transformada de Fourier con la herramienta

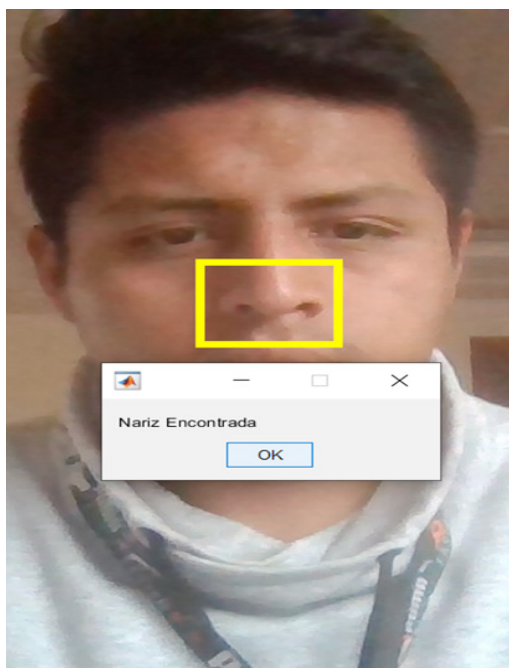
Matlab, la misma que nos arrojará una frecuencia definida para cada persona, la misma que se guardará en la base de datos para su posterior comparación.

La técnica consiste en que, al tener ya una base de datos con las frecuencias de los rostros de cada persona, simplemente es tomar la fotografía del rostro a reconocer, realizar el proceso de transformación de Fourier, obtener la frecuencia de esta imagen, acceder a la base de datos y realizar un proceso de comparación entre las diferentes frecuencias obtenidas. Al final, se realiza una simple resta de frecuencias y la que la más cercana esté a cero, constituirá el rostro de la persona a la que vamos a identificar.

Por otro lado, la técnica para reconocer si una persona está o no con su cubrebocas la detallamos a continuación:

La idea fundamental es primeramente tratar de reconocer que lo que nos muestra una imagen sea un rostro humano. Esto se fundamenta en que generalmente una mascarilla de bioseguridad cubre dos de los tres elementos fundamentales en un rostro humano: la nariz y la boca, siendo el tercero los ojos. En la siguiente figura 1, se muestra la detección de la nariz.

Figura 1 » Reconocimiento de nariz, uno de los tres elementos fundamentales de reconocimiento de un rostro humano.



Después de hacer que el algoritmo recorra la imagen y detecte los elementos descritos anteriormente se necesitaba que también lo haga con la forma de los rostros humanos ya que se lo programó para detectar todos los rostros humanos que aparezcan en la imagen ya sean estos de niños, adolescentes o adultos, tal como se puede evidenciar en la figura 2.

Figura 2 » Reconocimiento y detección de rostros humanos.



Nuestro algoritmo matemático reconoce estos tres elementos fundamentales en un rostro, el mismo que al no reconocer tanto la boca como la nariz (en este caso solamente los ojos), inmediatamente nuestro sistema infiere que la persona reconocida está con cubrebocas, tal como se muestra en la figura 3.

Figura 3 » Reconocimiento solo de ojos, por lo que el sistema infiere que está con mascarilla.



Por lo contrario, al reconocer tan solo uno de los tres elementos fundamentales como lo es la boca o nariz, el sistema infiere que no tiene colocado una mascarilla de bioseguridad, tal como se muestra en la figura 4.

Figura 4 » Reconocimiento solo de la boca, por lo que el sistema infiere que está sin mascarilla.



RESULTADOS

Una vez que hemos desarrollado el algoritmo en el software Matlab, realizamos las pruebas correspondientes para verificar la eficiencia del mismo, tomando en cuenta los distintos parámetros establecidos,

obteniendo datos de los distintos elementos del rostro humano para así poder determinar si las personas hacen uso de una mascarilla o no. Las pruebas se han realizado en distintos entornos y con distintos grupos de personas para determinar el margen de error que nos devuelve el programa y con esto realizar las mejoras para que el programa pueda darnos mejores resultados.

Cabe señalar que las pruebas se las ha realizado con el grupo de estudiantes de la materia de Matemáticas Avanzadas de la Carrera de Computación de la Universidad Politécnica Salesiana de la Sede Cuenca.

Nuestras pruebas también han dependido del entorno en el que se encontraba (aulas, casas, cantidad de luz, ambientes abiertos, etc.) y de la cámara que se estaba utilizando (cámara de celular conectada al computador, cámara externa, cámara interna).

A continuación, presentamos una tabla con los resultados obtenidos con distintos grupos de personas, en la que se especifica cuántas pruebas se han realizado.

Tabla 1: Resultados de las pruebas efectuadas con 5 diferentes grupos de personas

	1 persona	5 personas	8 personas	12 personas	25 personas
Número de pruebas	5	3	2	4	2
Aciertos	5	4	5	9	19
Falsos	0	1	3	3	6
Tasa de Acierto	100%	98%	75%	89%	94%

Como se puede apreciar en la Tabla 1, hemos realizado las pruebas con 5 grupos de personas con quienes probamos la eficiencia del sistema a través de un número de intentos y que se encuentran detallados, entregándonos un promedio de efectividad del 91%, lo cual muestra que nuestra aplicación tiene una alta eficiencia.

CONCLUSIONES

El software Matlab se considera eficiente al momento de hacer las comparaciones de las señales de los distintos puntos especificados ya que está precisamente diseñado para cálculos matemáticos muy avanzados.

Dicha eficiencia y capacidad de cálculo nos ha permitido detectar el rostro y otros puntos del rostro humano de una manera efectiva, ya que tiene un porcentaje bien bajo de errores al examinar la imagen. Cabe recalcar que

es tan eficiente a la hora de realizar estos experimentos ya que nos muestra los resultados de una forma muy lenta.

Matlab es muy eficiente en cálculos matemáticos, y no es un programa que se pueda transferir hacia otras máquinas o que funcione desde otra aplicación.

Los resultados obtenidos con este prototipo han sido favorables ya que se ha logrado detectar rostros conjuntamente con los elementos definidos (ojos, la nariz y la boca), elementos que coadyuvaron a definir si el usuario se encontraba con o sin mascarillas.

Al ser un programa funcional se le puede tomar como base para generar un proyecto más usable y portable.

La idea es mejorar su eficiencia utilizando otro lenguaje de programación (Java) el cual ayudaría a reducir el tiempo de respuesta de una manera considerablemente a la hora de detectar rostros.

BIBLIOGRAFÍA

- Caballero Barriga, E. R. (2017). Aplicación práctica de la visión artificial para el reconocimiento de rostros en una imagen, utilizando redes neuronales y algoritmos de reconocimiento de objetos de la biblioteca opencv.
- Daubechies, I. (1990). The Wavelet transform, time frequency localization and signal analysis, *IEEE Transactions on Information Theory*, 36(5), 961-1005.
- Díaz, A. B. M. (2004). Reconocimiento facial automático mediante técnicas de visión tridimensional (Doctoral dissertation, Universidad Politécnica de Madrid).
- Gallegos, M., Zalaquett, C., Luna Sánchez, S. E., Mazo-Zea, R., Ortiz-Torres, B., Penagos-Corzo, J., ... & López Miranda, R. (2020). Cómo afrontar la pandemia del Coronavirus (Covid-19) en las Américas: recomendaciones y líneas de acción sobre salud mental.
- Gualdrón O. E, Duque Suárez O. M, Chacón Rojas M. A. (2013, noviembre). Diseño de un sistema de reconocimiento de rostros mediante la hibridación de técnicas de reconocimiento de patrones, visión artificial e ia, enfocado a la seguridad e interacción robótica social. Universidad de Pamplona.
- Guillermo Mur. (2015). Inferencia de la Respuesta Afectiva de los Espectadores de un Video. Madrid: Universidad Carlos III de Madrid.
- Haro, A. S. (2020). Caracterización epidemiológica de Covid-19 en Ecuador. *InterAmerican Journal of Medicine and Health*, 3, 1-7.
- Jin, Z., Lou, Z., Yang, J., Sun, Q. (2007). Face detection and recognition unde nonuniform illumination. *Pattern Recognition*, 32, 1707-1718.
- Mur Igualada, G. (2015). Inferencia de la respuesta afectiva de los espectadores de un vídeo (Bachelor's thesis)
- Oracle, E. (s,f). Base de Datos Relacional. Oracle España
- Oracle, E. (s. f.). Base de Datos. Oracle España.
- Pérez, N. L., & Agudelo, J. T. (2012). Técnicas de biometría basadas en patrones faciales del ser humano (Doctoral dissertation, Universidad Tecnológica de Pereira. Facultad de Ingenierías Eléctrica, Electrónica, Física y Ciencias de la Computación. Ingeniería de Sistemas y Computación).
- Revelo Álava, A. S. (2019). Implementación de un algoritmo de detección de somnolencia humana, en tiempo real basado en visión artificial (Bachelor's thesis, Quito, 2019.).
- Revelo, A. (2019). Implementación de un algoritmo para la Detección de Somnolencia Humana, en Tiempo Real Basado en Visión Artificial. Quito: Escuela Politécnica Nacional.
- Valvert Gamboa, J. R. (2006). Métodos y Técnicas de Reconocimiento de Rostros en Imágenes Digitales Bidimensionales. Universidad de San Carlos de Guatemala.