

Algoritmos para el procesamiento de imágenes implementados en el Robot Humanoide InMoov.



Revista EIA
ISSN 1794-1237
e-ISSN 2463-0950
Año XVIII/ Volumen 18/ Edición N.36
Junio-Diciembre de 2021
Reia36019 pp. 1-16

Publicación científica semestral
Universidad EIA, Envigado, Colombia

PARA CITAR ESTE ARTÍCULO / TO REFERENCE THIS ARTICLE /

Valero Carvajal, D. A.; Orozco Isaza, J. E.; Córdoba Puerto, J. N.; Noreña Hemelberg, O. S.; Gómez Alarcón, M. A.; Jinete Gómez, M. A.; Gutiérrez Alfonso, J. A. (2021). Algoritmos para el procesamiento de imágenes implementados en el Robot Humanoide InMoov. Revista EIA, 18(36), Reia36019. pp. 1-16. <https://doi.org/10.24050/reia.v18i36.1495>

✉ Autor de correspondencia:

Valero Carvajal, D. A. (Diego Alejandro):
Pregado - Ingeniería mecatrónica.
Universidad Piloto de Colombia.
Correo electrónico:
diego-valero@upc.edu.co

Recibido: 17-01-2021

Aceptado: 15-05-2021

Disponible online: 01-06-2021

✉ **DIEGO ALEJANDRO VALERO CARVAJAL**¹

JUAN ESTEBAN OROZCO ISAZA¹

JAZMÍN NATALIA CÓRDOBA PUERTO¹

OSCAR SANTIAGO NOREÑA HEMELBERG¹

MIGUEL ALEJANDRO GÓMEZ ALARCÓN¹

MARCO ANTONIO JINETE GÓMEZ¹

JHONATAN ALEJANDRO GUTIÉRREZ ALFONSO¹

1. Universidad Piloto de Colombia.

Resumen

El mundo de la robótica parecía ser un mundo de ciencia ficción hasta hace pocas décadas, sin embargo, en los últimos años el desarrollo de nuevas tecnologías en los robots ha venido creciendo exponencialmente, hasta el punto en donde, estos se han incorporado en nuestras vidas, brindado apoyo en áreas tan diversas, como: la educación, la medicina, el transporte y la recreación, entre muchos otros escenarios, en donde el objetivo principal es desarrollar robots útiles y seguros, que buscan mejorar la calidad y la productividad de algunas actividades. En este artículo se presenta el desarrollo y la implementación de algoritmos de procesamiento digital de imágenes y machine learning, implementados en el robot humanoide InMoov, el primer robot open source fabricado a partir de impresión 3D en el mundo, desarrollado por el escultor francés Gaël Langevin en septiembre de 2011, estos algoritmos cuentan con funcionalidades para la detección y tracking de movimiento de personas, reconocimiento facial, control de la mano robótica del humanoide mediante el seguimiento de movimiento de la mano humana, además, se implementó en el robot, un sistema embebido sintetizador de voz, para que este sea capaz de realizar y responder algunas preguntas previamente definidas. Los algoritmos elaborados se implementan en el primer robot humanoide Inmoov ensamblado en Colombia, con estos se busca que el robot fortalezca la interacción humano-robot (HRI) y la autonomía en sus movimientos, los algoritmos de procesamiento de imágenes desarrollados cuentan con una eficiencia promedio del 90% de efectividad o acierto en ambientes con iluminación controlada.

Palabras Claves: Inmoov, Robot humanoide, Código abierto, Algoritmos, Robótica, Comunidad maker.

Algorithms for image processing implemented in the InMoov Humanoid Robot.

Abstract

The world of robotics seemed to be a science fiction world until a few decades ago, however, in recent years the development of new technologies in robots has been growing exponentially, to the point where they have been incorporated into our lives, provided support in areas as diverse as: education, medicine, transportation and recreation, among many other scenarios, where the main objective is to develop useful and safe robots that seek to improve the quality and productivity of some activities. This article presents the development and implementation of digital image processing and machine learning algorithms, implemented in the humanoid robot InMoov, the first open source robot manufactured from 3D printing in the world, developed by the French sculptor Gaël Langevin. In September 2011, these algorithms have functionalities for the detection and tracking of human movement, facial recognition, control of the humanoid's robotic hand by tracking the movement of the human hand, and a system was implemented in the robot embedded speech synthesizer, so that it is able to ask and answer some previously defined questions. The elaborated algorithms are implemented in the first Inmoov humanoid robot assembled in Colombia, with these it is sought that the robot strengthens the human-robot interaction (HRI) and autonomy in its movements, the developed image processing algorithms have an average efficiency 90% effectiveness or success in environments with controlled lighting.

Key Words: *Inmoov, Robot humanoide, Código abierto, Algoritmos, Robótica, Comunidad maker.*

Introducción

El desarrollo de robots humanoides, tiene como objetivo principal el imitar un determinado comportamiento humano, siendo uno de los campos involucrados el estudio de las tareas de interacción humano-robot (HRI), en donde como objetivos específicos destacan: el integrar funcionalidades empleando la visión por computadora, rama de la Inteligencia Artificial y el incorporar la gestión de diálogo (DM), áreas que han evidenciado un crecimiento exponencial durante los últimos años (Eidenmueller, 2017), prueba de ello, son los estudios realizados en diversos robots, tales como: ASIMO (Hirose & Ogawa, 2006), o ROBONAUT (Diftler, et al., 2003).

En esta investigación se toma como base el robot de código abierto InMoov Figura 1, este se implementa con los algoritmos de procesamiento de imagen para: detección de personas usando histogramas de gradientes orientados (HOG) y una máquina de soporte vectorial (SVM), reconocimiento facial por medio de entrenamiento de redes neuronales, mimesis de manos empleando la técnica envolvente convexa para el control de manos robóticas, además se implementa un sistema embebido sintetizador de voz, pregunta y respuesta. En el desarrollo del software se enfrentaron retos como implementar los algoritmos en un ordenador de tamaño reducido, teniendo en cuenta los factores: tamaño, peso, consumo y costo, al igual que las capacidades de procesamiento y la imprecisión de los sensores ópticos ante los cambios de iluminación, aspectos de vital importancia para este tipo de aplicaciones robóticas, con la imple-

mentación de estos algoritmos se busca que el robot sea interactivo con las personas y su medio de tal forma que pueda ser usado en aplicaciones para el aprendizaje, publicidad o entretenimiento.

Figura 1. Primer robot InMoov en Colombia.
Fuente: Elaboración propia.



Metodología y resultados

1.1 Algoritmo para la detección de movimiento y tracking de personas en tiempo real

Este algoritmo permite la detección de movimiento y el tracking de personas utilizando Python y apoyándose en la librería OpenCv, la cual es una herramienta open-source para visión artificial y machine learning. El código cuenta con un sistema para la detección previa del movimiento en tiempo real haciendo que el flujo de detección de personas solo se active si existe movimiento, así este ahorra recursos computacionales en el procesamiento ya que el algoritmo será probado en una Raspberry pi 3, para esta etapa de la implantación no se probara el algoritmo en ambientes exteriores. Una vez el flujo identifique una persona procede a realizar un cuadro delimitador sobre la persona reconocida, este proceso se describe en la Figura 2.

Figura 2. Esquema general de la implementación para la detección de personas. Fuente: Elaboración propia.

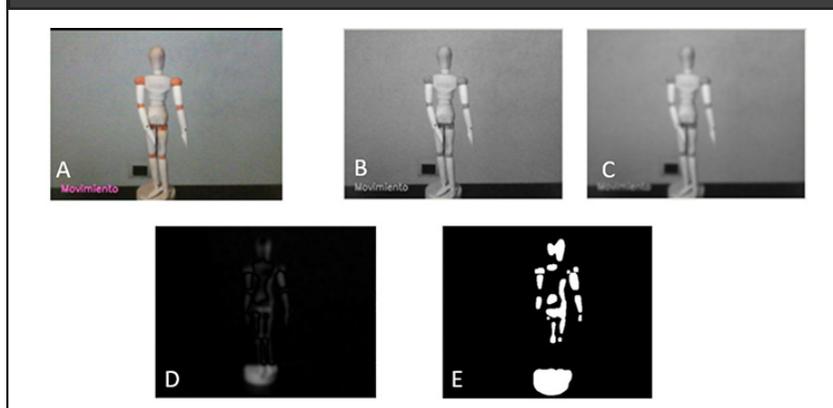


El algoritmo está dividido en dos partes principales; detección de movimiento y detección de personas, adicionalmente se divide en 5 etapas; adquisición, preprocesado, procesamiento y descripción e interpretación.

Adquisición: Es la primera etapa para el funcionamiento del algoritmo, en donde se importa la cámara y se obtienen los valores que se muestran en ella, aquí el tamaño de la imagen es definida por la resolución de la cámara, que para el caso es 640x480 píxeles, sin embargo, es necesario redimensionarla para trabajar con matrices más pequeñas definiendo un espacio de trabajo de 500 x 400 píxeles, así el proceso es más ligero computacionalmente (Szeliski, 2010), adicionalmente en esta etapa se toma y guarda una imagen de referencia Figura 3A, la cual es necesaria para el proceso posterior de detección de movimiento.

Preprocesado: En esta etapa se realiza una segmentación buscando eliminar información que no es necesaria para el procesamiento de la imagen de referencia, para ello se procede a convertir la imagen RGB a un solo canal en donde se toma el valor los tres canales (RGB) y se mezclan haciendo un ponderado para obtener tan solo un canal en escala de grises Figura 3B.

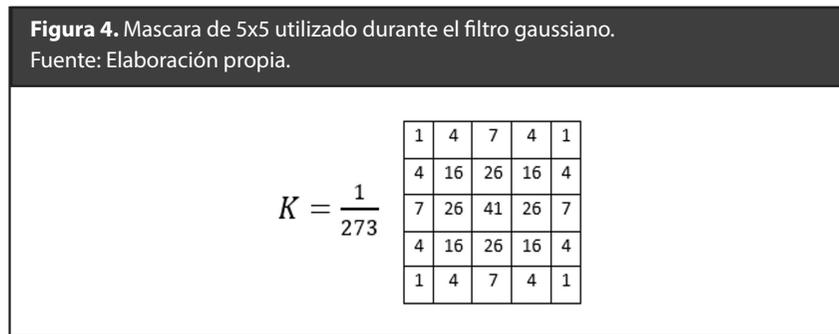
Figura 3. (A) Imagen original (B) Imagen en un solo canal (C) Filtro Gaussiano (D) Sustracción (E) Umbralización. Fuente: Elaboración propia.



Posteriormente se implementa un filtro gaussiano, este ayuda en el procesamiento para obtener mejores formas y reducir los componentes de alta frecuencia en la imagen actuando como un filtro pasa bajas (Deepthi & Sankaraiah, 2011), de esta manera la imagen se suaviza, esto se obtiene a partir de la función gaussiana presente en (1).

$$G(x, y) = e^{-\frac{x^2 + y^2}{2 * \sigma^2}} \quad (1)$$

Donde x es la distancia desde el origen en el eje horizontal, y es la distancia desde el origen en el eje vertical y σ es la desviación estándar de la distribución gaussiana, para el caso se usó una máscara de 5x5 con una desviación estándar $\sigma = 1$ que se evidencia en la Figura 4. Obteniendo como resultado la imagen de la Figura 3C, una vez que la máscara ha sido aplicada a todos los pixeles de la imagen.



Con la imagen suavizada obtenida se espera tres segundos para dar un tiempo en que pueda ocurrir movimiento, en esta etapa se obtienen los fotogramas de la cámara y se realiza el mismo preproceso mencionado anteriormente. Esta nueva imagen (img2) es comparada con la imagen referencia (img1) a través de una sustracción que retorna los valores absolutos de cada píxel representados en la Figura 3D empleando (2). La nueva imagen (Img2) se almacena y se renombra (Img1) de tal forma que, si el robot se moviese, aún pueda determinar si hay movimiento ya que la imagen base cambia.

$$Subtraction = Saturate (|Img1 - Img2|) \quad (2)$$

Luego de esta sustracción se realiza una binarización para representar la imagen con valores de 0 y 1, y una dilatación expresada mediante (3).

$$A \oplus B = \{x | (\hat{B}_x) \cap A \neq \emptyset\} \quad (3)$$

Donde A es la imagen restada y B es nuestro elemento estructural, que para el caso B=1, por ende, al recorrer la imagen con el elemento estructurante el tamaño de los objetos (región blanca) aumenta como se muestra en la Figura 2e.

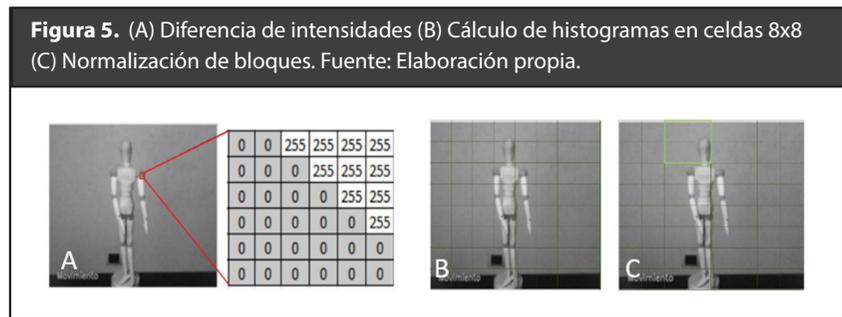
Segmentación: En esta etapa del proceso las imágenes ya están listas para ser procesadas, inicialmente se realiza una detección de contornos a las dos imágenes a partir del método Canny ya que este presenta un bajo costo computacional que facilita la implementación en tiempo real (Shopa, et al., 2014), para identificar si hay movimiento o no, es necesario obtener el área del contorno para compararlo con una

mínima área definida, de esta manera si supera el umbral se define que hay movimiento, mostrando en pantalla la palabra movimiento, adicional se procede con el reconocimiento de personas, para este proceso se aplica un histograma de gradientes (HOG), el cual ha demostrado que pueden clasificar con éxito hasta el 91% de los peatones en las áreas de atención (Bertozzi, et al., 2007), primero se calcula la diferencia de intensidad entre los pixeles vecinos a partir de (4) para las horizontales y (5) para las verticales.

$$dx = I(x + 1, y) - I(x - 1, y) \tag{4}$$

$$dy = I(x, y + 1) - I(x, y - 1) \tag{5}$$

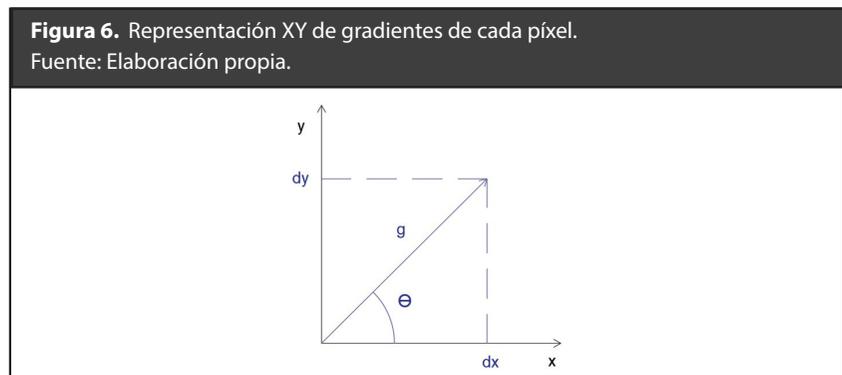
Obteniendo como resultado una matriz como la mostrada en la Figura 5A, la cual describe la imagen de manera numérica acotada en una escala de 8 bits.



Una vez se obtiene la matriz se calcula los gradientes de cada píxel definidos por la dirección donde el cambio de intensidad es máximo y la magnitud, por medio de (6) y (7), de esta manera el gradiente nos da información acerca de la forma del objeto.

$$\sigma(x, y) = \sqrt{dx^2 + dy^2} \tag{6}$$

$$g(x, y) = \arctan\left(\frac{dy}{dx}\right) \tag{7}$$



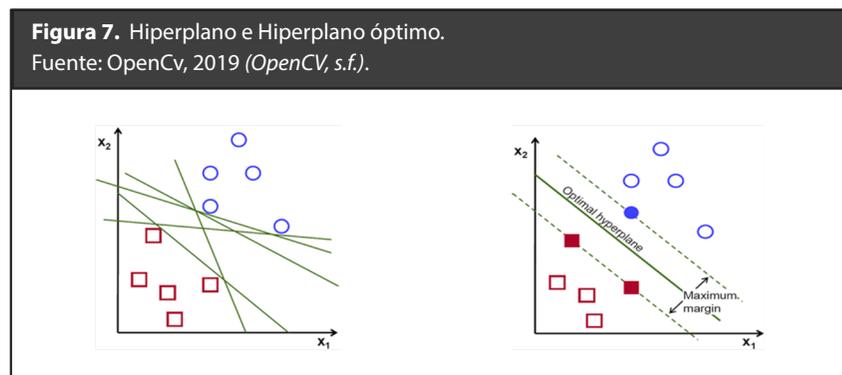
Se procede a convertir la información local del gradiente a una representación global en forma de vector de características usando el descriptor HOG, primero se divide la imagen en un número definido de celdas (8*8) Figura 5B, dependiendo de la dirección del gradiente obtenido mediante (7) se asigna a un intervalo definido para sumar todos los gradientes con ángulos similares por medio de (8), para crear el histograma en una posición k. Estos histogramas se realizan para cada celda.

$$h(k) = \sum_{(x,y) \in C} \omega_k(x,y)g(x,y) \quad (8)$$

Una vez se calcule el histograma para cada celda se realiza un proceso de normalización con una ventana deslizante de 2*2 celdas mostrada en la Figura 5C, la cual se encarga de concatenar los histogramas para que el descriptor sea más robusto a cambios de iluminación y ruido (Shu, et al., 2011) a partir de (9).

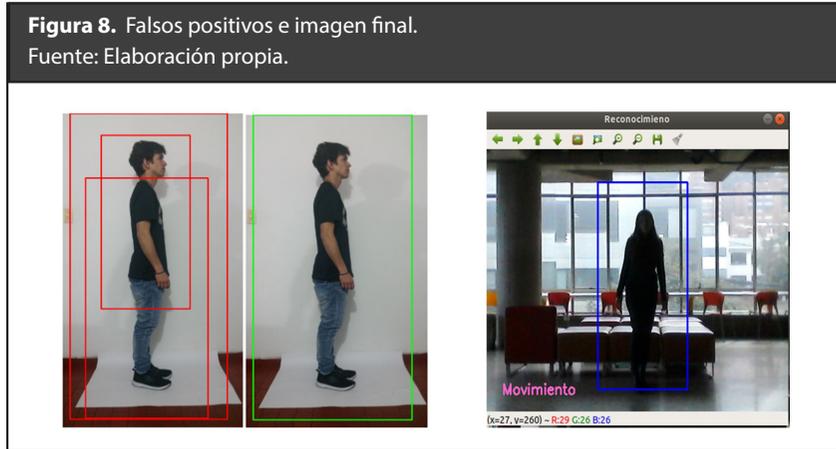
$$v' = \frac{v}{\sqrt{||v|| \frac{2}{2} + \epsilon}} \quad (9)$$

Esta información se utiliza en la máquina de soporte vectorial (SVM) para la detección y el tracking de las personas. El SVM tiene en cuenta los datos de entrenamiento etiquetados (aprendizaje supervisado), generando un hiperplano óptimo que clasifica nuevas muestras. La operación del algoritmo SVM se basa en encontrar el hiperplano que da la mayor distancia mínima a los ejemplos de entrenamiento. Así es como realiza la comparación para identificar que la imagen corresponde al objeto que se desea identificar.



Representación y descripción: En la etapa final del proceso se obtiene los cuadros delimitadores de salida del Histograma de gradientes, sin embargo, para eliminar los múltiples recuadros superpuestos (falsos positivos), producto del HOG, se usa una supresión no máxima, su función es tomar los cuadros delimitadores que se superponen y generar una relación entre 0 y 1 de superposición entre cuadros delimitadores, luego se aplica un umbral de 0.65 para determinar una superposición alta (cuadros similares), si la superposición supera el umbral, posteriormente se procede a eliminar los falsos positivos, dejando un único cuadro delimitador, el cual debería ser una persona como se muestra en la Figura 8. El algoritmo se implementó en un ordenador de tamaño reducido con un procesador de 64-bit a una frecuencia de 1.2 GHz y una memoria RAM de 1GB, haciendo que el programa tenga un rendimien-

to aproximadamente de 15 fotogramas por segundo (fps), además el algoritmo se probó con 20 personas que se desplazaban frente a la cámara a distancias entre 1 y 4 metros, bajo las mismas condiciones controladas de iluminación, donde se obtuvo un 95% de efectividad debido a que una persona no fue detectada correctamente.



1.2 Control de movimientos a partir de mimesis corporal empleando procesamiento de imágenes.

Esta funcionalidad permite al usuario mover las manos del robot, como si de sus propias manos se tratase, traduciendo cada una de las imágenes capturadas por la cámara en movimientos mecánicos precisos asemejándose a los realizados por el usuario. Este desarrollo se llevó a cabo mediante el tratamiento y procesamiento de imágenes, lo que involucra una serie de pasos y procesos los cuales se describirán a continuación.



Adquisición: La primera etapa del presente sistema de visión artificial consiste en la digitalización de la imagen. Para ello se utilizó una cámara digital con una resolución de 1280 x 800 píxeles, siendo esta el tamaño de la imagen a tratar y que se traduce en una función de luz bidimensional $f(x,y)$, donde X y Y indican las coordenadas espaciales y el valor de f en cualquier punto (x,y) siendo proporcional a la intensidad

luminosa, o nivel de gris de la imagen en dicho punto. En la Figura 9, se presenta el resultado de esta etapa de adquisición, la cual se representa mediante una matriz que contiene los valores a tratar en la subsecuente etapa.

Preprocesado: Se centra en hallar el contorno de la mano dentro de la imagen, con el objetivo de poder ubicarla y separarla de los demás objetos que están presentes en la escena, para ello se implementó una ecualización del histograma y se aplicaron filtros de tipo gaussiano y morfológico para la reducción del ruido. En primer lugar, se opta por trabajar con una región inferior seleccionando una pequeña área de interés (ROI) dentro de la imagen a tratar, así mismo se realiza la segmentación del tono de la piel, pasando del modelo de color RGB (Red, Green, Blue) al modelo HSV (Hue, Saturation, Value), haciéndolo más robusto ante los cambios de iluminación (Pralhad Salunkhe & Anil Patil, 2015). Sin embargo, es necesario normalizar el valor de este modelo a una escala de 8 bits, que abarque valores de 0 a 255, debido a que el modelo HSV trabaja con ángulos, comprendidos entre 0° a 360° .

Segmentación: Involucra la aplicación de un filtro gaussiano de 5×5 píxeles Figura 4, con el objetivo de suavizar la imagen y eliminar algún tipo de ruido que pueda afectar el proceso, además se define el rango de valores de cada uno de los canales del modelo HSV, con la intención de establecer el color deseado, los cuales se tendrán en cuenta en la etapa de umbralización, que contempla la binarización de cada uno de los píxeles presentes en la imagen, en donde aquellos píxeles que se encuentren fuera del rango toman un valor de 0 (negro) y los que se encuentran contenidos, toman el valor de 1 (blanco) usando (10), donde $src(x,y)$ representa la imagen de entrada, $dst(x,y)$ representa la imagen de salida y $thresh$ es el valor del umbral.

$$dst(x,y) = \begin{cases} maxVal & \text{if } src(x,y) > thresh \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

Con la imagen binarizada, se procede a la eliminación de ruido aplicando un filtro morfológico, con el objetivo de reducir o, en el mejor de los casos, eliminar ciertos píxeles que no hacen parte de la estructura de la mano y pueden alterar los resultados, el filtro aplicado consta de una erosión seguido de una dilatación que, en síntesis, consiste en una apertura, regida por (11).

$$A \circ K = (A \ominus K) \oplus K \quad (11)$$

Finalmente, se obtiene como resultado una nueva imagen que contempla, únicamente los píxeles que contienen el valor establecido como se muestra en la Figura 10.

Figura 10. Entrada vs Salida, aplicando la umbralización.
Fuente: Elaboración propia.

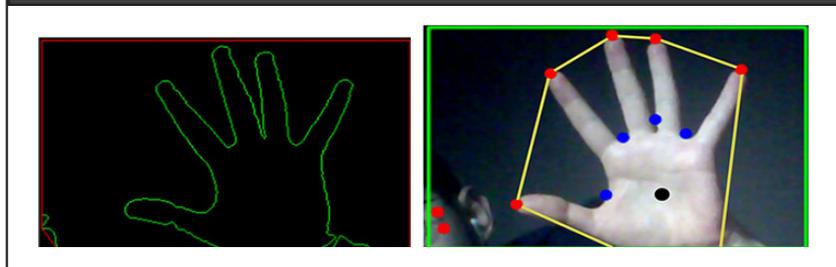


Una vez obtenido este resultado se procede a ubicar los contornos presentes en la escena a partir de la imagen binarizada, seguido del cálculo de la técnica denominada Envoltente Convexa, la cual consiste en realizar la operación de un polígono cuyos lados se obtienen a partir de la unión de los puntos más externos del contorno, eliminando de esta manera las concavidades y defectos de convexidad. Para esta técnica se utilizó la fórmula Envoltente Convexa mostrada en (12) (Buitrago, et al., 2015).

$$C(x) = \left\{ \sum_{i=1}^k \alpha_i x_i \mid x_i \in X, x_i \in R, \alpha_i \geq 0, \sum_{i=1}^k \alpha_i = 1 \right\} \quad (12)$$

Dado la cantidad de n puntos dentro del contorno de la imagen binarizada x_1, x_2, \dots, x_n , donde su envoltente será la intersección de todos los contenidos involucrados. Así mismo, los resultados obtenidos mediante permite identificar los puntos mínimos y máximos locales, es decir, el extremo de cada uno de los dedos y sus concavidades.

Figura 11. Resultante al aplicar la Envoltente Convexa, obteniendo los puntos máximos (rojos), mínimos (azules) y negro (centroide del polígono).
Fuente: Elaboración propia.



Representación y descripción: A la salida del proceso de segmentación obtenemos los datos de la cantidad de puntos máximos presentes en la escena, excluyendo

todos aquellos que tengan un ángulo superior a 90 grados entre sí o una distancia mayor a 25 píxeles a partir el centroide del polígono, mediante (13) se puede calcular esta distancia, determinando que los valores que se encuentran fuera de dicho límite no hacen parte de los dedos de la mano, contando únicamente con los puntos que formen parte del área establecida (mano) como se observa en la Figura 11. Cabe destacar que los requisitos anteriormente mencionados fueron establecidos a partir de las distintas pruebas realizadas con anterioridad, estableciendo que la mano debe encontrarse dentro de la zona de interés (ROI) delimitada por un recuadro de color verde, así mismo se determinó que la mano no debe encontrarse a una distancia mayor de 45 cm de la cámara para que el sistema continúe teniendo valides. De igual modo, la técnica empleada generalmente devuelve más puntos de los que se necesitan, por lo que existe la necesidad de filtrarlos, para ello todos los puntos que se encuentren a una distancia inferior de 5 píxeles entre sí son promediados con el objetivo de dejar tan solo un punto, además solo se contarán con un máximo de 5 dedos, por lo tanto, si el sistema identificara un número mayor a este automáticamente serían ignorados. Donde $p_1 = (x_1, y_1)$ son los puntos máximos de la Envolvente Convexa y $p_2 = (x_2, y_2)$ es el punto del centroide de la mano.

$$d(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_2 - y_1)^2} \quad (13)$$

Por último, los valores son almacenados en una variable tipo contador, correspondientes al número de dedos de la mano, dato que será enviado a la siguiente fase del proceso.

Control de la mano robótica

La información obtenida del anterior proceso es enviada a través de la red, empleando el protocolo de comunicación TCP IP, basado en IoT, que es recibida por el sistema embebido principal del InMoov, el cual ejecuta las siguientes instrucciones:

Preparación: Este primer apartado crea un servidor al cual llegarán los datos, ubicado dentro del sistema embebido con funcionalidad wi-fi, presente en el robot, para tal fin se define la dirección IP y su puerto de conexión (Sen Gupta, et al., 2009), el cual se convierte en el servidor al que se conectará el cliente, en este caso, el ordenador de tamaño reducido responsable de enviar los datos resultantes del procesamiento de imágenes; Así mismo es necesario inicializar cada uno de los servomotores involucrados en el movimiento de las manos y muñeca, siguiendo los parámetros de la caracterización presentes en la Tabla 1 y Tabla 2, en donde los valores máximos y mínimos corresponden a los grados de apertura y cierre para cada uno de los dedos y las muñecas.

Tabla 1. Rango de ángulos de movimiento de cada dedo, en la mano izquierda.
Fuente: Elaboración propia.

M. Izquierda	Mínimo [°]	Inicial [°]	Máximo [°]
Pulgar	0	5	180
Índice	40	18	180
Corazón	0	0	150
Anular	30	150	150
Meñique	20	20	180
Muñeca	50	50	140

Tabla 2. Rango de ángulos de movimiento de cada dedo, en la mano derecha.
Fuente: Elaboración propia.

M. Derecha	Mínimo [°]	Inicial [°]	Máximo [°]
Pulgar	0	0	180
Índice	0	0	180
Corazón	0	0	180
Anular	0	0	180
Meñique	40	45	180
Muñeca	50	50	140

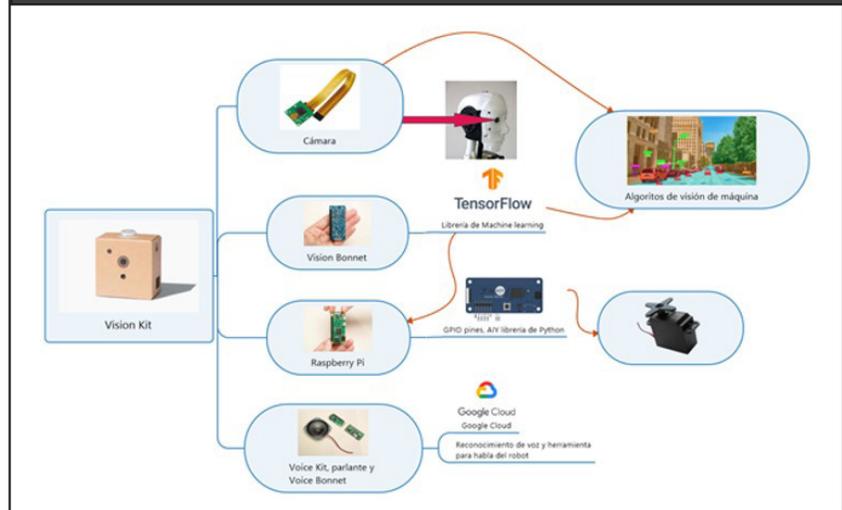
Conectividad: para garantizar la seguridad de los datos se implementó el protocolo de seguridad IPV4, con protección por contraseña WPA2 para redes de internet inalámbricas, de esta manera solo se podrán conectar los usuarios y dispositivos autorizados (Gonzalez Paz, et al., 2016).

Control de actuadores: corresponde al algoritmo que contiene los parámetros e instrucciones correspondientes al movimiento individual de cada servomotor, según corresponda los valores recibidos determinan la posición de cada dedo.

1.3 Reconocimiento de objetos por medio de redes neuronales y reconocimiento de voz por medio de librerías de google assistant

Este algoritmo permite al robot InMoov interactuar con su entorno como se muestra en la Figura 12, mediante sensores ópticos, micrófonos y altavoces, utilizando los kits de desarrollo de visión de máquina y reconocimiento de voz desarrollados por la compañía Google, estos kits utilizan un ordenador de tamaño reducido con la capacidad suficiente para procesar todos los datos involucrados. El kit de visión de máquina utilizado es la Vision Bonnet equipada con un chip especialmente diseñado para correr algoritmos de machine learning, de igual manera en el kit se implementó la librería aiy de Python, la cual permite utilizar modelos de machine learning pre desarrollados con la librería TensorFlow, otorgando la capacidad de enviar señales de control a los actuadores del robot.

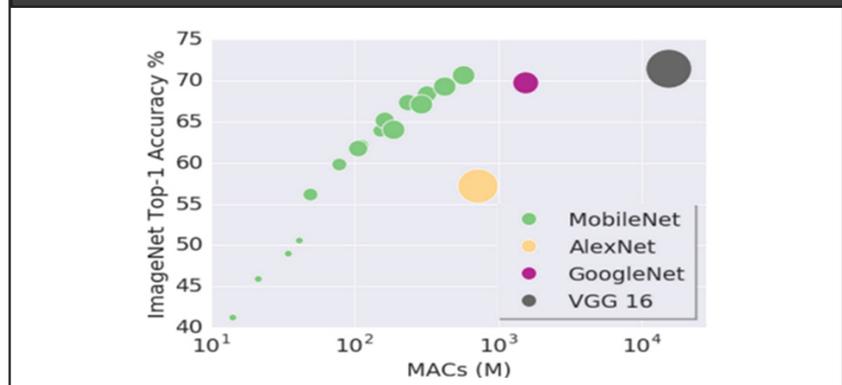
Figura 12. Kit de visión de máquina y reconocimiento de voz implementados en el robot InMoov. Fuente: Elaboración propia.



El kit de reconocimiento de voz cuenta con un ordenador de tamaño reducido, una placa Voice Bonnet, la cual incorpora dos micrófonos, un conector para altavoz y un procesador códec de audio, elementos que trabajan en sinergia junto a las herramientas Google Cloud y Google Assistant siendo la voz del InMoov, además se incorporó la funcionalidad de interactuar con el servomotor encargado de mover la mandíbula para su gesticulación vocal.

Para entrenar la red neuronal con el objetivo de que el InMoov detecte objetos y personas, primero se instaló la versión 1.7 de la librería TensorFlow (Xie & Guo, 2019) y se utilizó una base de datos de imágenes elaborada para el proyecto, la cual contiene las caras de sus integrantes para que el InMoov pueda reconocerlos con una red neuronal llamada MobileNet, esta es una red neuronal convolucional pequeña y eficiente, convolucional significa que los cálculos se hacen para cada píxel, o locación de la imagen. La red neuronal se reentrenó utilizando la configuración recomendada de imágenes preprocesadas de 224 píxeles. Se seleccionó esta red neuronal por su eficiencia ya que esta necesita hacer menos cálculos para tener un buen porcentaje de efectividad.

Figura 13. Comparación de redes neuronales. Fuente: Google TensorFlow (Google TensorFlow, 2018).



La Figura 13 muestra la precisión de la configuración (eje y), frente al número de cálculos necesarios (eje x) y el tamaño del modelo (área del círculo), comparado con otras redes neuronales.



La red neuronal se sometió a diferentes pruebas buscando que detectara los rostros de las personas previamente entrenadas con diferentes escenarios de iluminación, en donde de 100 muestras detecto correctamente 88 de ellas, esto demuestra las ventajas de usar una red neuronal convolucionada.

Conclusiones

En el desarrollo de la investigación se presentaron varios problemas en el procesamiento de imagen debido a los costos computacionales ya que se implementaron en un sistema de bajas prestaciones computacionales, sin embargo, este ofrece un bajo costo económico y un tamaño ideal para las dimensiones del robot, además los escenarios de iluminación debieron ser controlados para obtener una detección más precisa, para ello cada desarrollo cuenta con diferentes filtros y métodos de preprocesado para que los algoritmos sean lo más robustos ante cambios de escenarios y eficientes.

En las pruebas realizadas para el algoritmo de detección y tracking de personas se obtiene una eficiencia del 95% cuando el escenario tiene un ambiente controlado, además se evidencia que el detector HOG y el SVM fueron configurados con los parámetros adecuados para el requerimiento. La razón del 5% de error es cuando una persona se desplaza rápidamente por la cámara, de esta manera el algoritmo no suprime los falsos positivos correctamente. Como trabajo futuro se espera robustecer el algoritmo de tal forma que este pueda ser usado en ambientes con iluminación alta, baja o de intermitencia.

El algoritmo para la detección de la mano y el seguimiento de los movimientos de la misma, demuestra que la técnica envolvente convexa tiene resultados satisfactorios al momento de aislar los puntos de interés correspondientes a cada dedo de la mano, sin embargo, en ambientes con iluminación no controlada los resultados variaron debido a que el valor del umbral fue establecido para trabajar en un ambiente específico con un tipo de luz determinada, ocasionando que el sistema encontrará más falsos positivos que interfirieren con el proceso. Para trabajos futuros se debe tener en cuenta que, al momento de segmentar, todo aquello que se asemeje al tono de piel hará parte del contorno.

Implementando los kits de desarrollo Google se obtiene el hardware necesario para utilizar algoritmos de reconocimiento de objetos y reconocimiento de voz, al utilizar librerías de Google y sus servicios podemos desarrollar algoritmos robustos con herramientas y funciones preestablecidas obteniendo muy buenos resultados. Se comprobó que la red neuronal MobileNet tiene precisión de la configuración (eje y), frente al número de cálculos necesarios (eje x) y el tamaño del modelo (área del círculo), lo cual beneficia la eficiencia del algoritmo obteniendo un 88% de efectividad en ambientes no controlados.

Los algoritmos implementados dan cumplimiento al objetivo general, fortaleciendo la interacción humano robot (HRI) y la autonomía del robot, ya que el InMoov tiene la capacidad interactuar con personas, reconociendo e identificando al sujeto, siguiendo parte de sus movimientos y respondiendo algunas de sus preguntas.

Referencias

- Bertozzi, M., Broggi, A., Del Rose, M., Felisa, M., Rakotomamonjy, A., & Suard, F., (2007). IEEE Intelligent Transportation Systems Conference, 30 Sept - 3 Oct 2007, Seattle: USA, A Pedestrian Detector Using Histograms of Oriented Gradients and a Support Vector Machine Classifier, IEEE Xplore, 22 Oct 2007. Disponible en: <https://doi.org/10.1109/ITSC.2007.4357692>
- Buitrago, O., Ramírez, A. & Britto, R. (2015). Nuevo Algoritmo para la Construcción de la Envoltura Convexa en el Plano. Información tecnológica, 26 (4), pp 1-9. <https://doi.org/10.4067/S0718-07642015000400017>.
- Deepthi, R. S., & Sankaraiah, S., (2011). IEEE Conference on Open Systems (ICOS2011), 25-28 Sept 2011, Langkawi: Malaysia, Implementation of Mobile Platform Using Qt and Open CV For Image Processing Applications, IEEE Xplore, 15 Nov 2011. Disponible en: <https://doi.org/10.1109/ICOS.2011.6079235>
- Diftler, M., Culbert, C. J., Ambrose, R. O., Plat, R., & Bluethmann, W. J., (2003). IEEE International Conference on Robotics and Automation, 14-19 Sept 2003, Taipei: Taiwan, Evolution of the NASA/DARPA Robonaut control system, IEEE Xplore, 10 Nov 2003. Disponible en: <https://doi.org/10.1109/robot.2003.1241975>
- Eidenmueller, H. (2017). The Rise of Robots and the Law of Humans. Oxford Legal Studies Research Paper No. 27/2017, pp 2-8. <https://doi.org/10.2139/ssrn.2941001>
- Gonzalez-Paz, A., Beltran-Casanova, D., & Fuentes-Gari, E. R. (2016). PROPUESTA DE PROTOCOLOS DE SEGURIDAD PARA LA RED INALÁMBRICA LOCAL DE LA UNIVERSIDAD DE CIENFUEGOS. Universidad y Sociedad. 8(4), pp 130-137.
- Google TensorFlow. (2018), TensorFlow For Poets. [En línea]. Disponible en: <https://codelabs.developers.google.com/codelabs/tensorflow-for-poets/#3> (Último acceso: 26 August 2020)
- Hirose, M., & Ogawa, K. (2006). Honda humanoid robots development. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 365(1850), pp 11-19. <https://doi.org/10.1098/rsta.2006.1917>
- OpenCV, s.f. (2018), Docs.opencv.org. [En línea]. Disponible en: https://docs.opencv.org/3.4/d1/d73/tutorial_introduction_to_svm.html (Último acceso: 26 August 2020)
- Pralhad Salunkhe, R. & Anil-Patil, A., (2015). Third International Conference on Image Information Processing (ICIIP), 21-24 Dec 2015, Wagnaghat: India, Image processing for mango ripening stage detection: RGB and HSV method, IEEE Xplore, 25 Feb 2016. Disponible en: <https://doi.org/10.1109/ICIIP.2015.7414796>
- Sen Gupta, G., Finnie, M. & Mukhopadhyay, S., (2009). IEEE Instrumentation and Measurement Technology Conference, 5-7 May 2009, Singapore:Singapore, WiFi-based control of a robotic arm with remote vision, IEEE Xplore, 21 Jul 2009. Disponible en: <https://doi.org/10.1109/IMTC.2009.5168512>

- Shopa, P., Sumitha, N. & Patra, P. S. K., (2014). International Conference on Information Communication and Embedded Systems (ICICES2014), 27-28 Feb 2014, Chennai:India, Traffic Sign Detection and Recognition Using OpenCV, IEEE Xplore, 9 Feb 2015. Disponible en: <https://doi.org/10.1109/ICICES.2014.7033810>
- Shu, C., Ding, X., & Fang, C. (2011). Histogram of the Oriented Gradient for Face Recognition. Tsinghua Science and Technology 16(2), pp 216-224. [https://doi.org/10.1016/S1007-0214\(11\)70032-3](https://doi.org/10.1016/S1007-0214(11)70032-3)
- Szeliski, R. (2010). Computer Vision: Algorithms and Applications. Springer-Verlag London Limited 2011. Springer, London. Disponible en <https://doi.org/10.1007/978-1-84882-935-0>
- Xie, L. & Guo, X., (2019). IEEE International Conference on Smart Internet of Things (SmartIoT), 9-11 Aug 2019, Tianjin: China, Object Detection and Analysis of Human Body Postures Based on TensorFlow. Tianjin, China, China, IEEE Xplore, 14 Nov 2019. Disponible en: <https://doi.org/10.1109/SmartIoT.2019.00070>