

DISEÑO DE UN ROBOT CARTESIANO DE UN EJE PARA LA MEDICIÓN DE TEMPERATURA CORPORAL Y DETECCIÓN DE MASCARILLA UTILIZANDO SENSORES INFRARROJOS Y VISIÓN ARTIFICIAL

GUTIERREZ SANCHEZ Sergio, HERNÁNDEZ VALLADARES, Mario, MAGALLANES CHAUCA Diego

*Escuela Profesional de Ingeniería
Mecatrónica, Universidad Ricardo Palma,
Lima, Perú*

sergio.gutierrez@urp.edu.pe

mario.hernandez@urp.edu.pe

diego.magallanes@urp.edu.pe

Resumen — Este proyecto tiene la finalidad de desarrollar un Robot Cartesiano que va a ayudar en el control de acceso de los compradores en los distintos centros comerciales en todo Lima. Este Robot Cartesiano será instalado en las entradas de los centros comerciales, en donde va a medir la altura del cliente mediante un sensor de ultrasonido para luego elevar el cartesiano a un 80% de la altura total del cliente para poder detectar, utilizando una cámara, si los clientes portan o no una mascarilla además de medir la temperatura del cliente mediante un sensor de infrarrojo; si tienen una temperatura aceptable y porta la mascarilla, se procederá a abrir la puerta del centro comercial para permitirle el acceso.

Palabras claves – Robot cartesiano; visión artificial; sensor infrarrojo; Covid-19

Abstract — This project has the purpose of developing a Cartesian Robot that will help in the control of access of the buyers in the different shopping centers throughout Lima. This Cartesian Robot will be installed at the entrances of the shopping centers, where it will measure the height of the client by means of an ultrasound sensor and then raise the Cartesian to 80% of the total height of the client to be able to detect, using a camera, whether or not customers wear a mask in addition to measuring the customer's temperature using an infrared sensor; If they have an acceptable temperature and you are wearing the mask, the door of the shopping center will be opened to allow you access.

Keywords - Cartesian Robot; convolutional neural networks; artificial vision; infrared sensor; Covid-19

I. INTRODUCCIÓN

Debido a la pandemia, las empresas se ven obligados a contratar personal adicional para que esté presente en los ingresos de los establecimientos comerciales con el objetivo de medir la temperatura y permitir el acceso al lugar. Esto causa un exceso mensual en los costos de la empresa para pagar a este personal y además de ponerlos en riesgo debido a la constante interacción con otras personas.

Por ello, se buscó como solución diseñar un robot cartesiano de 1 eje que usando un sensor ultrasónico obtenga la altura de

la persona que desea ingresar al centro comercial y en consecuencia controle la altura del sensor para que el cliente coloque su muñeca para su posterior medición de temperatura. También se encargará, por medio de una cámara, de tomar una foto del cliente para verificar si porta una mascarilla o no. Y en caso se encuentre en el rango permitido y porte una mascarilla, se le autorizará el ingreso al centro comercial, abriendo la puerta de manera automática; en caso contrario, le avisará que podría tener algún problema de salud. Si por algún motivo el cliente no quiere cumplir estas indicaciones, se procederá a pasar por medio de una alarma a las autoridades del establecimiento para que tomen cartas en el asunto. Todo esto para mantener la bioseguridad en los ingresos de los centros comerciales y evitar poner a los clientes en riesgo.

II. MARCO TEORICO

A. Centro Comercial

Los centros comerciales nacen con el objetivo de atraer un gran número de clientes que busquen satisfacer una necesidad de consumo, ya sea adquiriendo artículos específicos o realizando compras de impulso. La ventaja de estas edificaciones es que, al congregarse un gran número de comercios, el cliente tiene más alternativas entre las cuales elegir. Todo esto conlleva a que este sea uno de los lugares con más números de clientes en cualquier día.

B. Robot Cartesiano

Estos tipos de robots funcionan con sus 3 ejes principales de manera lineal con el fin de formar ángulos rectos unos respecto a otros. Por otra parte, también es el robot más fácil de concebir y fabricar, como se ve en la Figura 1, porque su producción es tan sencilla como ir acoplando múltiples actuadores lineales unos sobre otros perpendicularmente. Indudablemente también es una de las construcciones más económicas.

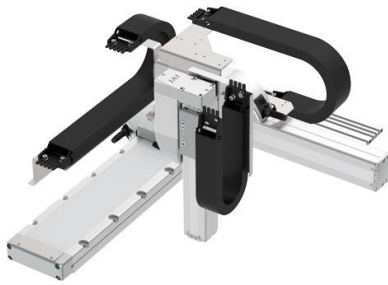


Fig. 1 Robot Cartesiano de 2 ejes

C. Sensor Ultrasonido (HC-SR04)

Este es un sensor de distancia que utiliza ultrasonido para determinar la distancia de un objeto en un rango de 2 a 450 cm. Destaca por su pequeño tamaño, bajo consumo energético y buena precisión, como se aprecia en la Figura 2. El sensor HC-SR04 es el más utilizado dentro de los sensores de tipo ultrasonido, principalmente por la cantidad de información y proyectos disponibles en la web. De igual forma es el más empleado en proyectos de robótica como robots laberinto o sumo, y en proyectos de automatización como sistemas de medición de nivel o distancia.



Fig. 2 Sensor Ultrasonido HC-SR04

D. Sensor de Temperatura (MLX90614)

Este sensor de temperatura infrarrojo permite medir la temperatura de un objeto a distancia, sin necesidad de tener contacto. El Sensor MLX90614 es un chip de silicio con una fina membrana micro mecanizada, como se logra apreciar en la Figura 3, diseñada para ser sensible a la radiación infrarroja emitida por un objeto a distancia. La salida del sensor es lineal y se compensa de acuerdo a las variaciones de la temperatura ambiente. Esta entrega un amplio rango de trabajo para objetos desde -70°C hasta 380°C , con una precisión de 0.5°C .



Fig. 3 P160 - ZKTeco

E. GOPRO hero 7 Black

Este es el modelo de cámaras de tamaño compacto, con una dimensión de $62,3 \times 44,9 \times 33$ mm, como se ve en la Figura 4, para instalaciones en mecanismos robóticos, con una resolución de foto de 30fps, y de video de hasta 4k a 60fps.



Fig. 4 GoPro hero 7 black

F. Motor Paso a Paso (Nema 23)

Los motores paso a paso son ideales en todas aquellas aplicaciones donde se necesita de fuerza al mismo tiempo que precisión. Este motor unipolar tiene una fuerza de 9Kg/cm con un diámetro de eje de 6mm y en formato NEMA 23. Es un motor bipolar con 4 cables de control. Usa unos 2A por fase por lo tanto necesitas un controlador potente.



Fig. 5 Motor Nema 23

G. MEDIAPIPE FACE DETECTION

Es una solución de detección de rostros ultrarrápida que viene con 6 puntos de referencia y soporte para múltiples rostros. Se basa en BlazeFace, un detector facial liviano y de buen rendimiento diseñado para inferencias de GPU móviles. El rendimiento en tiempo real del detector permite que se aplique a cualquier experiencia de visor en vivo que requiera una región facial de interés precisa como entrada para otros modelos de tareas específicas, como el punto clave facial 3D o la estimación de geometría, clasificación de rasgos o expresiones faciales y segmentación de la región facial.

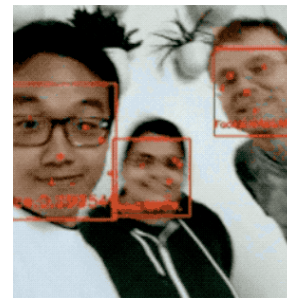


Fig. 6 Face Detection de MediaPipe

H. REDES NEURONALES CONVOLUCIONALES

Es un tipo de red neuronal artificial con aprendizaje supervisado que busca imitar al cortex visual del ojo humano con la finalidad de identificar distintas características de las entradas. Para ello, contiene varias capas ocultas especializadas y con una jerarquía, esto quiere decir que las primeras capas pueden detectar líneas, curvas y se van especializando hasta llegar a capas más profundas que reconocen formas complejas. Son muy efectivas para tareas de visión artificial, como en la clasificación y segmentación de imágenes.

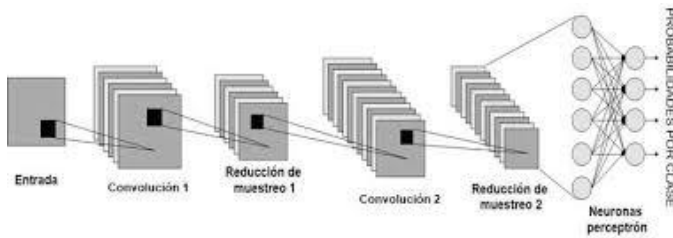


Fig. 7 Estructura de una Red Neuronal Convolutional

I. OPENCV

Conocido como Open Source Computer Vision es una librería software open-source de visión artificial y machine learning, esta librería tiene más de 2500 algoritmos, que permiten identificar objetos, caras, clasificar acciones humanas en vídeo, hacer tracking de movimientos de objetos, extraer modelos 3D, encontrar imágenes similares, seguir el movimiento de los ojos, etc. Se usa en aplicaciones como la detección de intrusos en vídeos, monitorización de equipamientos, ayuda a navegación de robots, inspeccionar etiquetas en productos, etc.



Fig. 8 Reconocimientos de imágenes y videos

J. MOBILENET

Es un modelo de visión capaz de distinguir entre 1000 clases de objetos diferentes, diseñados específicamente para ser ejecutados de manera eficiente en dispositivos móviles e integrados. Este se basa en una arquitectura optimizada y utiliza convoluciones separables profundas para construir redes neuronales profundas y livianas.

III. SISTEMA DEL ROBOT CARTESIANO

A. Diseño del Robot Cartesiano en SolidWorks

Este robot realizará su desplazamiento en el eje Z como se observa en la figura 9 con la finalidad de mover el módulo de medición y va a ser controlado por una Raspberry Pi en función al 80% de la altura de la persona medida con un sensor de ultrasonido ubicado en el techo de la entrada del centro comercial. Después de realizar el desplazamiento, utilizando el módulo de medición se realizará la toma de temperatura corporal mediante un sensor infrarrojo y mediante una cámara GOPRO Hero 7 Black se detectará si el cliente porta o no una mascarilla; si cumple los requisitos se procederá a abrir las puertas del centro comercial para que ingrese.

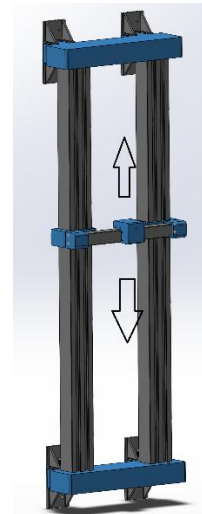


Fig. 9 Diseño en SolidWorks

Con la señal que el sensor de proximidad mida, este hará que los motores paso a paso, instalados como se aprecia en la Figura 10, que estarán conectados a los tornillos sin fin mediante un acoplador, y estos girarán hasta que el sensor infrarrojo este a la altura requerida.

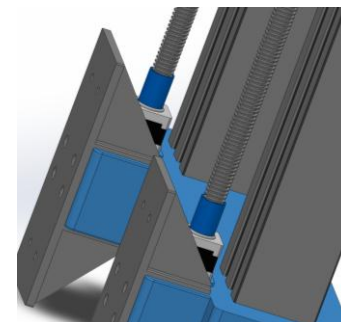


Fig. 10 Motores paso a paso de 50x50mm

B. Dimensionado del Robot Cartesiano

El robot diseñado en SolidWorks, cuenta con una altura de poco más de 2 metros, esto con la finalidad de que pueda hacer su funcionamiento con cualquier persona, sean jóvenes altos o niños. Todas las medidas se aprecian en la figura 11.

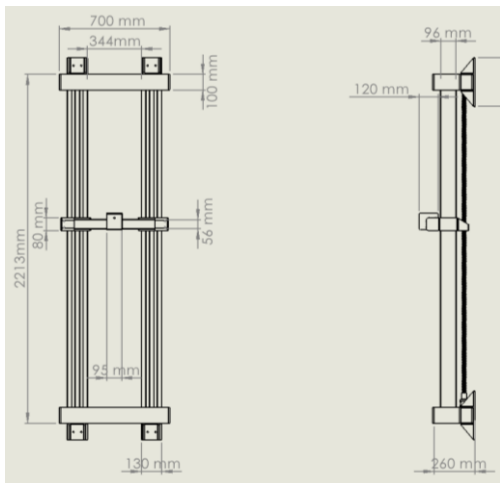
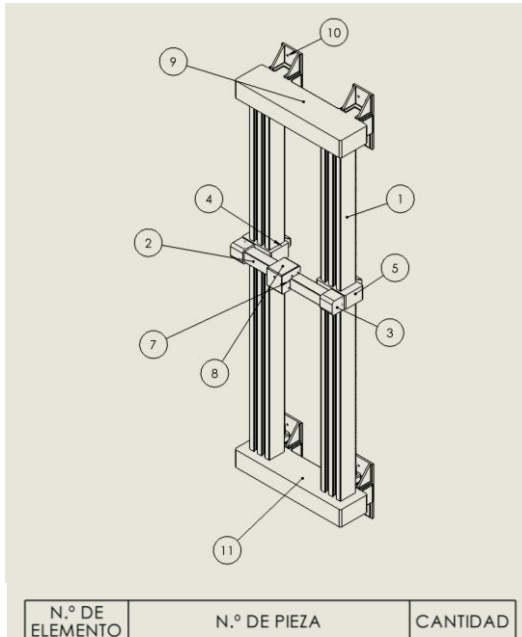


Fig. 11 Dimensionado del Robot Cartesiano

C. Partes del Ensamblaje del robot cartesiano

Este tipo de robot consta de más de 10 partes sin contar los tornillos para su ensamblaje total, como se observa en la figura 12.



N.º DE ELEMENTO	N.º DE PIEZA	CANTIDAD
1	Barra	2
2	Barra Horizontal	1
3	SopORTE barra vertical 2	2
4	SopORTE de barra vertical 1 Der	1
5	SopORTE de barra vertical 1 Iza	1
6	Tornillo sin fin	2
7	Tapa de caja	1
8	Caja	1
9	Parte Superior	1
10	SopORTE	4
11	Base	1
12	Motor paso a paso	2
13	Acople	2

Fig. 12 Indicación de partes para el Ensamblado

D. Análisis Dinámico usando el método de Lagrange-Euler

Para desarrollar este método, debemos realizar el algoritmo de Denavi-Hartenberg para hallar la cinemática directa del robot.

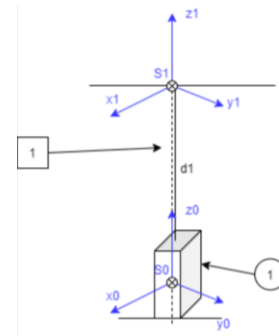


Fig. 13 Esquema del robot cartesiano

TABLA I. PARÁMETROS CINEMÁTICOS

Articulación	θ_i	d_i	a_i	α_i
1	0	d_1	0	0

Donde:

d_1 = desplazamiento variable del robot cartesiano

Luego se realiza el cálculo de la dinámica utilizando el método de Lagrange-Euler y se obtiene:

$$F_1 = m_1 * (\ddot{d}_1 + 9.8)$$

Donde:

F_1 = fuerza ejercita por el motor

m_1 = masa de la caja con la electrónica

\ddot{d}_1 = segunda derivada del desplazamiento variable del robot cartesiano

E. Simulación del Circuito para el Funcionamiento en Tinkercad

Para la simulación práctica del circuito electrónico se utilizó Tinkercad, una plataforma de diseño para modelado 3D y circuitos electrónicos perteneciente a Autodesk. Se ejecuta en la web y es reconocido por su simplicidad y facilidad de uso.

Como se observa en la figura 14, el circuito está siendo simulado en la web para su posterior implementación en el robot cartesiano.

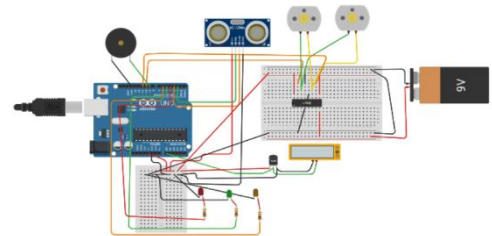


Fig. 14 Indicación de partes para el Ensamblado

Al detectar a una persona, el sistema se activa midiendo su tamaño para luego posicionarse a la altura correcta, de lo

contrario el sistema dará un mensaje de que no hay nadie. Posteriormente se medirá la temperatura y se calculará si la medición de la persona es aceptable o no lo es. De serlo una luz verde y un sonido indicarán que la persona está saludable, caso contrario la luz será roja. En la figura 15 se observa parte del código usado para la simulación en la web.

```

1 // Definir las variables a utilizar
2 const int echo = 3;
3 const int trigger = 2;
4 float tiempo, distancia;
5
6 const int led_rojo = 4;
7 const int led_verde = 5;
8 const int led_naranja = 6;
9
10 int tempPin = A0;
11 float temperatura = 0.0;
12
13 #define motorizq 10
14 #define motorder 11
15
16 const int pinbuzzer = 9;
17
18 void setup(){
19 // PREPARAR LA COMUNICACION SERIAL
20 Serial.begin(9600);
21
22 // CONFIGURAR PINES DE ENTRADA Y SALIDA
23 pinMode(echo, INPUT);
24 pinMode(trigger, OUTPUT);
25
26 pinMode(led_rojo, OUTPUT);
27 pinMode(led_verde, OUTPUT);
28 pinMode(led_naranja, OUTPUT);
29
30 pinMode(10, OUTPUT);
31 pinMode(11, OUTPUT);
32
33 pinMode(pinbuzzer, OUTPUT);
34 }
35
36 void loop() {
37 Serial.print("\nNuevo Ciclo");
38 delay(2000);
39 digitalWrite(led_rojo,LOW);
40 digitalWrite(led_verde, LOW);
41 digitalWrite(led_naranja, LOW);
42 // SI HAY PRESENCIA SE ACTIVA EL MECANISMO, DE LO CONTRARIO
43 // se desactiva

```

Fig. 15 Código utilizado para la simulación del sistema en Tinkercad

F. Diseño del Sistema de Detección de Mascarillas

El desarrollo del sistema de detección de mascarillas se realizó en dos etapas: la detección de rostros y la detección de mascarillas. En la primera etapa, utilizando OpenCV se captura en tiempo real el video proveniente de la cámara y mediante el módulo de Face Detection de MediaPipe se procede a obtener el rostro de las personas que se encuentren en el campo de visión y se indicará mediante un recuadro el rostro detectado como se observa en la figura 16, además, la imagen dentro del recuadro se almacenará aparte para ser procesada en la etapa de detección de mascarillas.



Fig. 16 Detección de rostro usando MediaPipe

En la segunda etapa, se procedió a reentrenar la red neuronal convolucional MobileNet debido a que es un modelo eficiente y ligero por lo que se podrá ejecutar con pocos recursos en la Raspberry Pi. Para la detección de mascarillas se modificó la capa de entrada para que pueda procesar una imagen de 128x128 píxeles en formato RGB y la capa de salida para indicar si cuenta o no con mascarilla utilizando la función de activación Sigmoidal. Para el entrenamiento se utilizaron 12800 imágenes y para la validación se utilizaron 790 imágenes, las cuales fueron obtenidas de Kaggle y se pre procesaron al tamaño y formato concordante a la capa de entrada de la red neuronal convolucional. Se obtuvo un 99.82% de precisión como máximo con los datos de entrenamiento y un 100% precisión como máximo con los datos de validación los cuales se observan en la figura 17.

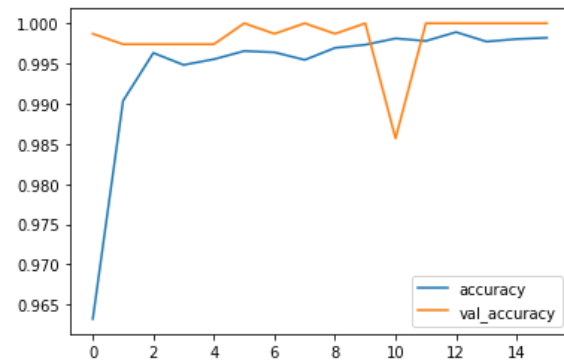


Fig. 17 Gráfico de precisión vs. Épocas del entrenamiento

Luego del entrenamiento, se procedió a implementar el modelo obtenido en el Sistema de Detección de Mascarillas. Las imágenes de los rostros obtenidos con MediaPipe son pre procesadas para cumplir con las condiciones de la capa de entrada de la red neuronal convolucional para luego identificar la presencia o no de las mascarillas, lo cual se va a indicar mediante un mensaje en la visualización de la imagen que va a cambiar entre el color rojo y verde indicando la ausencia o presencia de mascarilla respectivamente como se observa en la figura 18 y 19, además, va a enviar el resultado al sistema principal del Robot Cartesiano para proceder con permitir o no el acceso dependiendo de la temperatura corporal y si cuenta o no con mascarilla.



Fig. 18 Detección de persona sin mascarilla



Fig. 19 Detección de persona con mascarilla

G. Simulación en CoppeliaSim

Para realizar la simulación en el software CoppeliaSim se realizó la importación de los archivos stl del ensamblaje; luego se procedió a realizar la programación del sensor ultrasónico y de los motores para que cuando detecte a la persona, el sensor de ultrasonido mida su altura y luego se desplace el cartesiano al 80% de la medida obtenida para que luego realice la medición de la temperatura y la comprobación de la presencia de la mascarilla.

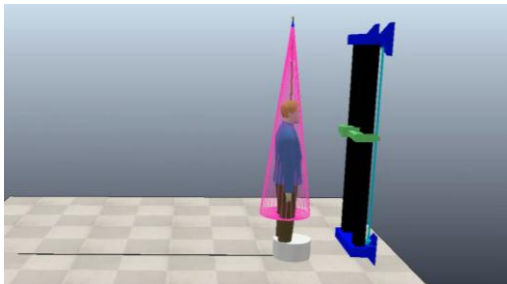


Fig. 20 Simulación en CoppeliaSim

IV. CONCLUSIONES

- En la parte del diseño, se tuvo en consideración a diversos tipos de soportes. Primero se pensó en empotrarlo al suelo, pero debido a su altura de casi 2mtrs y su masa de aproximadamente 200kg sería muy inestable y podría caerse, por lo que se optó en empotrarlo a la pared con 4 soportes instalados en cada esquina.
- En el diseño electrónico, se tomó en cuenta la variedad de motores y sensores a utilizar, usando aquellos más cercanos a los que se vayan a utilizar en la realidad para tener un primer prototipo del funcionamiento del sistema.
- Se observó durante las pruebas del sistema de detección de mascarillas que cuando una persona se tapa la boca con las manos o con algún objeto diferente a una mascarilla el sistema detecta que si tiene mascarilla por lo que es necesario agregar imágenes de personas tapándose la cara en el dataset para que el sistema sea capaz de identificar estos casos adecuadamente.
- En la simulación con CoppeliaSim, se tuvo que agregar un filtro a la medida obtenida con el sensor ultrasónico para que el movimiento del cartesiano sea suave.

REFERENCIAS

- [1] Mechatronics, N. (2018, 1 febrero). *Sensor de ultrasonido HC-SR04*. <https://naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html>
- [2] Mechatronics, N. (2018, 21 febrero). *Sensor de temperatura infrarrojo*. <https://naylampmechatronics.com/sensores-temperatura-y-humedad/330-sensor-de-temperatura-mlx90614.html>
- [3] Pujara, A. (2020, 14 junio). *Image Classification With MobileNet*. Analytics Vidhya. <https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470>
- [4] Aprende Machine Learning. (s.f.) *¿Cómo funcionan las Convolutional Neural Networks? Visión por Ordenador*. <https://www.aprendemachinelarning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/>
- [5] Kaggle. (s.f.). *Face Mask Detection 12k Images Dataset*. <https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset?select=Face+Mask+Dataset>
- [6] Kaggle. (s.f.). *Face Mask Detection Dataset*. <https://www.kaggle.com/omkargurav/face-mask-dataset>
- [7] MediaPipe. (s.f.). *MediaPipe Face Detection*. https://google.github.io/mediapipe/solutions/face_detection
- [8] OpenCV. (s.f.) *OpenCV modules*. <https://docs.opencv.org/4.5.3/>