UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL CARRERA DE

ELECTRÓNICA Y AUTOMATIZACIÓN

‘Implementación de Reconocimiento Facial y Visión artificial en Robot Nao con Python y OpenCV’

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO ELECTRÓNICO

AUTOR:

- **ISAAC GIOVANNI RIVERA ITURBURU**
- **DOUGLAS FABRICIO ZAMBRANO GUARANDA**

TUTOR: ING. DIEGO FREIRE

GUAYAQUIL – ECUADOR

2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Douglas Fabricio Zambrano Guaranda con documento de identificación N° 0923923569 e Isaac Giovanni Rivera Iturburu con documento de identificación N° 0952392595; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera totalo parcial el presente trabajo de titulación.

Guayaquil, 19 de febrero de 2022

Atentamente,

Douglas Z

Douglas Fabricio Zambrano Guaranda
C.I. 0923923569

Isaac Rivera

Isaac Giovanni Rivera Iturburu
C.I.0952392595

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Douglas Fabricio Zambrano Guaranda con documento de identificación No. 0923923569 e Isaac Giovanni Rivera Iturburu con documento de identificación No. 0952392595,expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Artículo Académico: Implementación de Reconocimiento Facial y Visión artificial en Robot Nao con Python y OpenCV, el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 19 de febrero de 2022

Atentamente,

Douglas Z

Douglas Fabricio Zambrano Guaranda
C.I. 0923923569

Isaac Rivera

Isaac Giovanni Rivera Iturburu
C.I. 0952392595

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Diego Roberto Freire Quiroga con documento de identificación N°0917208084 , docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: Implementación de Reconocimiento Facial y Visión artificial en Robot Nao con Python y OpenCV, realizado por Douglas Fabricio Zambrano Guaranda con documento de identificaciónN° 0923923569 y por Isaac Giovanni Rivera Iturburu con documento de identificación N° 0952392595 , obteniendo como resultado final el trabajo de titulación bajo la opción que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 19 de febrero de 2022

Atentamente,

Ing. Diego Roberto Freire Quiroga
C.I. 0917208084

DEDICATORIA

Dedico este Artículo Académico a mis padres, José Zambrano y Patricia Guaranda, por apoyarme todo este tiempo en mi vida universitaria y profesional; y con mucho cariño, trabajo y esfuerzo lograron convertirme en el ser humano que soy hoy en día.

A mi novia, quien me apoyó y me alentó a seguir adelante, para no rendirme y lograr ser un gran profesional.

Dedico este logro a todos mis familiares y a personas en general, que me han apoyado en el proceso de avance y culminación de este Académico Científico para poder finalizar con total éxito y satisfacción.

Douglas Fabricio Zambrano Guaranda

Dedico este artículo científico a mis padres, Freddy y Marion, quienes me han apoyado siempre y me han guiado con valores para ser una persona de bien y poder lograr las metas que me proponga en mi camino.

A mi hermana y a mis demás familiares que me han dado su apoyo a lo largo de mi vida y en micarrera universitaria.

Isaac Giovanni Rivera Iturburu

AGRADECIMIENTO

Ante todo, le agradezco a Dios, por darme salud e inteligencia para poder cumplir con mis metas propuestas y por haberme acompañado y guiado a lo largo de mi vida Universitaria y Profesional, para seguir adelante y no rendirme jamás.

Agradezco a mis padres, por darme apoyo en todo momento y haberme dado la oportunidad de tener una excelente educación, que me ayudó a sobresalir y cumplir con todas mis metas.

Agradezco a mi compañero de tesis y amigo, por ayudarme en el transcurso de mi carrera Universitaria y el tener buen trabajo en equipo para conseguir este Artículo Académico con total éxito.

Agradezco a los profesores de la Universidad Politécnica Salesiana, en especial a nuestro tutor el Ing. Diego Freire, que con su ayuda y su correcto asesoramiento pudimos lograr la finalización de este Artículo Académico con satisfacción.

Douglas Fabricio Zambrano Guaranda

Isaac Giovanni Rivera Iturburu

RESUMEN

Este artículo académico tuvo como finalidad implementar un reconocimiento facial y visión artificial en el robot humanoide Nao, mediante el lenguaje de programación Python y la librería Open CV, para darle una utilización más óptima al robot Nao, con el que cuenta la Universidad Politécnica Salesiana sede Guayaquil, el cual ha sido poco utilizado teniendo en cuenta que este artefacto tiene múltiples capacidades en que podrían ser de gran apoyo en los distintos procesos que comprende una universidad como organización. Un caso puntual es la acción de reconocer expresiones y a partir de esa información programar actividades de respuesta para el público. Para dicho objetivo, se pretende importar las librerías a utilizar, entre cuales, las más importantes son: 'Pynaoqi', la librería de OpenCV ('cv2'), el algoritmo Haar Cascade y la librería Numpy. Como proceso de metodología se utilizó el desarrollo de un programa en el software Pycharm, mediante el lenguaje de programación Python y usando las librerías de Open CV para lograr el principal objetivo del reconocimiento artificial y una comunicación estable por parte del Robot Nao que se encuentra disponible en la Universidad Politécnica Salesiana.

El desarrollo de este proyecto inició con la creación de una carpeta donde se guardó las fotos de los rostros de las personas que estarán en una base de datos desde diferentes ángulos para que el Robot pueda reconocer y procesar las imágenes de los rostros con ayuda de las librerías mencionadas anteriormente; para esto se desarrolló un algoritmo en el lenguaje de programación Python, por medio del cual va almacenar la biometría del rostro humano y sus nombres, para luego verificar expresiones faciales y programar respuestas. Los resultados de esta investigación, mostraron que Nao podría tener múltiples funciones, tanto de reconocimiento facial, como reconocimiento de voz y de expresiones. La facilidad de este lenguaje de programación permitió que el funcionamiento de Nao fuera entendible, aún para personas que no son expertos en programación. Para las pruebas finales realizadas para el reconocimiento de monedas, se evidenció que para mayor efectividad en el reconocimiento facial, el robot precisa de una mayor resolución en sus cámaras.

PALABRAS CLAVE: Visión Artificial, Reconocimiento Facial, Open CV, Python, Redes Neuronales, Nao, Inteligencia Artificial, Algoritmo Haar Cascade.

ABSTRACT

The purpose of this academic article was to implement facial recognition and artificial vision in the Nao humanoid robot, using the Python programming language and the Open CV library, to give a more optimal use to the Nao robot, which has been little used considering that this artifact has multiple capabilities that could be of great support in the various processes that comprise a university as an organization. A specific case is the action of recognizing expressions and from that information to program response activities for the public. In this sense, we intend to import the libraries to be used, among which the most important are: 'Pynaoqi', the OpenCV library ('cv2'), the Haar Cascade algorithm and the Numpy library. The methodology used was the development of a program in PyCharm software, using the Python programming language and the Open CV libraries to achieve the main objective of artificial recognition and the stable communication by the Nao Robot available at the Salesian Polytechnic University.

The development of this project began with the creation of a folder where the photos of the faces of the people that will be in a database, were saved in different angles so that the Robot can recognize and process the images of the faces with the help of the libraries mentioned above; and finally the algorithm was developed in the Python programming language, through which the biometrics of the human face and their names will be stored, to then verify facial expressions and program responses. The results of this research showed that the Nao Robot could have multiple functions, both facial recognition and voice and expression recognition. The facility of this programming language allowed the operation of Nao to be understandable, even for people who are not experts in programming. For the final tests performed for the recognition of coins, showed that for greater effectiveness in facial recognition, the robot requires a higher resolution in its cameras.

Keywords: Artificial Vision, Facial Recognition, Open CV, Python, Neural Networks, Nao, Artificial Intelligence, Haar Cascade Algorithm.

INDICE DE CONTENIDO

INTRODUCCIÓN

El reconocimiento facial y métrica de los rostros constituyen una parte fundamental de diversas aplicaciones que toman como referencias los rostros humanos. El auge de la visualización de esquemas faciales, alineados con factores como la iluminación, genera que la tarea de reconocimiento de rostros en el mundo real sea un reto.
En ese sentido, se precisan de diversos métodos de reconocimiento facial, como por ejemplo la red neuronal, la cual podría definirse como un mecanismo para hallar esa combinación de parámetros y hacer aplicación a la vez.

Dentro de los lenguajes de programación existentes, se encuentran Java, Python, Matlab y C++. Estas son herramientas fundamentales en el desarrollo de esta implementación, considerando que Java, es un lenguaje popular que se ejecuta en la Máquina Virtual Java (JVM). A su vez, permite una portabilidad perfecta entre plataformas, sin embargo, su principal limitante se encuentra en la capacidad de análisis de datos.
Por su parte, Python es una excelente opción de lenguaje para el estudio de datos, y no solo en el nivel de entrada. Gran parte del proceso de ciencia de datos gira en torno al proceso ETL (extracción-transformación-carga). Es muy emocionante para el Aprendizaje Automático (Machine Learning). Asimismo, Matlab es un lenguaje de computación numérica que se utiliza en el mundo académico y en la industria. La utilización de este podría ser ese en el cual tu aplicación necesite una función matemática intensa y avanzada.
Por último, C++, es un lenguaje de programación que surge de la extensión del lenguaje C, y su funcionalidad se da enla manipulación de objetos.

Una vez observado los beneficios de estos lenguajes, se determinó utilizar Python, ya que este cuenta con una gran variedad de uso, entre ella la inteligencia artificial, lo que permite adaptarla idea principal de este estudio que es el Robot Nao, para el reconocimiento facial. Asimismo, Python será el lenguaje por medio del cual se analizará y tabularán los datos obtenidos del reconocimiento facial.

PROBLEMAS DE ESTUDIO

El desarrollo tecnológico a nivel mundial cada vez va en aumento, principalmente en términos de inteligencia artificial y robótica como factores fundamentales en la automatización de procesos dentro de las organizaciones. Es el caso puntual del uso de los Robots Nao, lo definencomo un sistema tipo humanoide, que tiene la capacidad artificial de interactuar con las personas, posee la capacidad de hablar, escuchar y ver, según sea su programación [1]

En distintas áreas de la cotidianidad humana el uso de la tecnología es inminente, más cuando se le da un enfoque que ayude a reducir los errores y las restricciones que se evidencian en la mano de obra humana. Es así como el uso de la robótica y la inteligencia artificial ha contribuido a descongestionar grandes volúmenes de datos y dar soluciones a problemáticas extensas como por ejemplo se podría destacar la lucha contra la pobreza extrema y la contribución a mejorar la calidad de vida en áreas remotas de distintas formas, debido a que ayuda a identificar las causas de la pobreza y a detectar las regiones más necesitadas.

Lo anterior solo es por mencionar algunos de los beneficios que esta herramienta puede ofrecer. Puntualmente el tema del reconocimiento facial está en pleno auge de la cotidianidad, considerando que distintas plataformas como Facebook que hace reconocimiento facial a través de fotografías identificando a las personas a través de sus bases de datos o los dispositivos celulares que tienen la capacidad artificial de desbloquearse con solo mirarlo. Es por esta razón que diversas organizaciones están migrando a procesos complejos que ocasionan equivocaciones en distintos momentos al uso de la robótica y la inteligencia artificial. Este algoritmo lo que realiza es una toma de medidas biométricas del rostro como la boca, los ojos y la nariz para determinar su reconocimiento a través de mediciones de anchura de ojos o distancia de la nariz y la boca, entre otros rasgos.

Algunas organizaciones han optado por la utilización del robot Nao, no solo como herramienta de reconocimiento facial, sino como medio de interacción con las personas, en función de receptar información física y emocional para luego dar un diagnóstico sobre

su estado final. Actualmente, la Universidad Politécnica Salesiana cuenta con el robot humanoide Nao, lo cual representa una ventaja competitiva en términos de innovación. Sin embargo, su utilización no ha sido de gran provecho, considerando que esta herramienta tiene múltiples capacidades en términos de automatización, como, por ejemplo, el reconocimiento facial y visión artificial. Estees un tema que se ha implementado muy poco, por lo que surge la necesidad de realizar una investigación que permita mostrar una faceta más de esta herramienta desde la carrera de Electrónica y Automatización.

Calvopiña & Valladares [2], afirman que este sistema que se puede ir adaptando según la necesidad de quien lo utiliza, utilizando software de alto nivel como Choreographe y Python. Se define este software como “un tipo de lenguaje de scripting independiente de plataforma y encaminado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web” [3], La finalidad de este estudio, va permitir desarrollar una nueva faceta del Robot Nao para la Universidad, enfocado en el reconocimiento facial de las personas que ingresan y salen del establecimiento, a través del reconocimiento biométrico de rostros de distintas personas, para obtener datos que se guardarán en las bases para uso de la Universidad.

JUSTIFICACIÓN

Esta investigación tiene como finalidad la implementación de un reconocimiento facial y visión artificial con el robot Nao, mediante Python y Open CV. Uno de las principales necesidades que se busca solucionar con esta investigación es incentivar a los diferentes grupos de interés, especialmente los estudiantes en formación a fortalecer sus habilidades en cuanto al uso de la Robótica, de manera que se pueda presentar exposiciones, seminarios y casa abierta donde puedan encontrar en la carrera de Electrónica y Automatización su futura profesión y un nichode aprendizaje con aplicaciones en visión y reconocimiento artificial usando el robot Nao.

También, se debe considerar que la automatización de procesos está teniendo gran impacto en las distintas organizaciones, como factor competitivo y de desarrollo. Específicamente este estudio está realizado en la Universidad Politécnica Salesiana sede Guayaquil en la carrera de Electrónica y Automatización se usaría al robot humanoide Nao como objeto de prácticas de laboratorio en la materia de Robótica Móvil, con esto permitirá a los estudiantes ver una de las muchas aplicaciones que se podrían realizar con el robot Nao.

OBJETIVO GENERAL

Implementar un reconocimiento facial y visión artificial en el robot humanoide Nao mediante Python y Open CV.

OBJETIVOS ESPECÍFICOS

- Crear una base de datos con el robot Nao para el reconocimiento facial.
- Desarrollar un código con el lenguaje de programación Python y la biblioteca Open CV en base al algoritmo de identificación y clasificación de rostros ‘Haar Cascade’ que permitirá el reconocimiento facial y la visión artificial por parte del Robot humanoide Nao.
- Permitir que el robot Nao tenga un pequeño diálogo con las personas, las cuales tengamos las imágenes de sus rostros guardadas en la base de datos de Nao.
- Aplicar la visión artificial para tener reconocimiento de monedas por parte del Robot Nao.

MARCO TEÓRICO REFERENCIAL

El robot Nao es un robot humanoide de 58 cm, es una herramienta interactiva y programable, que está en constante evolución. Tiene la capacidad de interactuar con cualquier tipo de personas de una forma natural. Responde a una programación enfocada en cualquier sentido, ya sea ver dialogar u oír. En cuanto a sus movimientos no hay limitantes, ya que este Robot, podría jugar un partido de futbol, dirigir y realizar promoción de un producto o servicio [1].

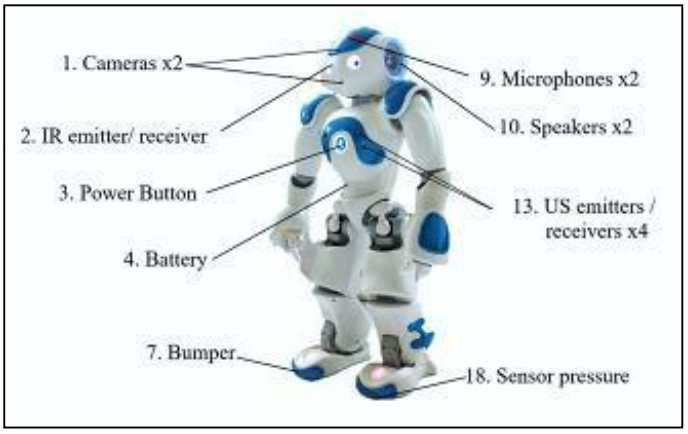


Fig. 1. Descripción Técnica del Robot Nao [3]

Nao, percibe el entorno a partir de sus múltiples sensores, los cuales comprenden: dos cámaras, cuatro micrófonos, nueve sensores táctiles, dos sensores de ultrasonido, 8 sensores de presión, un acelerómetro y un giroscopio. Además, incluye un software de programación gráfico llamado Choreographe, el cual da viabilidad para la programación, sin que se tenga conocimientos de un lenguaje de programación. Para los usuarios expertos, comprende un conjunto completo para desarrollo de software, que permite usar distintos lenguajes como: C++,Python, JAVA, .NET y MATLAB.

“Una práctica que se pudo observar con el Robot fue una terapia para niños con trastornos de autodesarrollo desarrollada por la Universidad Tecnológico de Monterrey, este trabajo denominado ‘Nao Juega Conmigo’ la cual se trató de una acción enfocada a niños autistas, enla cual se hizo un enfoque de aplicar modelos de aprendizaje para niños, que precisan desarrollar la habilidad de lenguaje, esto como prueba de que Nao, puede ejecutar este tipo de terapias” [4]. Por otra parte, la robótica es la tecnología que se encarga de diseñar, construir y operar un robot, esta área de la robótica se caracteriza por la

complejidad de las diversas áreas que se manejan ahí, estas son [2]:

- ✓ Mecatrónica
- ✓ Sistemas Electrónicos
- ✓ Sistemas de Cómputo
- ✓ Hardware
- ✓ Software

Asimismo, existen diversos tipos de robots, con distintos modos de aplicación, según la necesidad de las personas. Estos tipos de se pueden clasificar según Cortés [5] como se muestra a continuación:

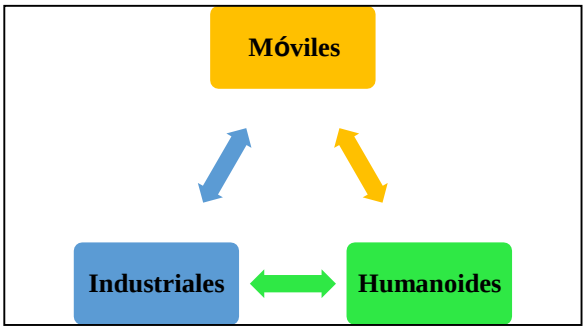


Fig. 2. Clasificación de los robots [5]

INTELIGENCIA ARTIFICIAL. La Inteligencia Artificial se define, como un conjunto de algoritmossecuénciales, que siguen una ruta orientada por una máquina, generalmente estos pasos, no tienen la característica de ser lineales, por el contrario, son eventuales; es decir sucederá dependiendo la necesidad y de esa manera actuará. Es decir, que la finalidad de estos algoritmos es poder consolidarse en máquinas tecnológicas que sean capaces de cumplir múltiples tareas como los humanos [6].

La inteligencia artificial en el Robot Nao se refiere a que este tenga la capacidad de reconocera las personas que se someten a la prueba de reconocimiento facial, a través de sus rasgos enel rostro, con un conocimiento previo que se constituye en una base de datos que le permita hacer comparaciones que se traducen en resultados óptimos.

REDES NEURONALES. Por otra parte, las redes neuronales se constituyen como una técnica que proviene de la Inteligencia Artificial, considerando que busca encontrar o proponer soluciones informáticas, imitando la forma en la que actúa el cerebro humano. Cabe resaltar que la capacidad del cerebro humano de pensar, recordar y resolver problemas ha captado de la atención de científicos a que traten de imitar el comportamiento del cerebro de una persona [7]. La capacidad que muestra Nao es muy similar al cerebro humano, ya que utiliza la base de datos como recuerdos o conocimientos, para luego arrojar un resultado a partir de la detección facial.

MACHINE LEARNING. Se podría considerar como un componente científico del escenario de la Inteligencia Artificial, que tiene como finalidad crear sistemas que desarrollan aprendizaje automáticamente. Cuando se refiere a aprender dentro de este modelo, tiene significado de identificación de patrones de alta complejidad en millones de datos. El Machine Learning en otras palabras con el diseño de nuevas máquinas que emulan la inteligencia humana, busca suplir la necesidad de algunas actividades que realizan los humanos y a su vez, reducir el margen de error [8].

PYTHON. "Se considera como un lenguaje de programación, el cual interpreta por su naturaleza la legibilidad de su código. Por medio de Python, se puede hacer diversas acciones que van a permitir que Nao en este caso genere sus propios Scripts, a través de los cuales se podría indicar al robot directrices para que actúe según se programe, tanto desde un terminal Python como desde las cajas de Choreographe" [9]. Así también, Martelli [10], afirma que una característica de este sistema de programación es su versatilidad, además de su rápido aprendizaje y amplio contenido, lo que permite que se adapte de forma fácil. El autor también menciona algunas ventajas y desventajas que tiene esta herramienta:

Ventajas

- Es un sistema de código abierto
- Multiplataforma
- Está encaminado a objetos
- Facilidad para su aprendizaje

Desventajas

- Interpretación un poco más lenta de procesar que un lenguaje compilado.

Python, es el lenguaje que utiliza Nao para mostrar los resultados posteriores al reconocimiento facial, este lenguaje de programación a la facilidad de utilizar al robot según sea la necesidad y su lenguaje es de fácil entendimiento para los que son principiantes en el mundo de la programación.

OPEN CV. Se podría definir como biblioteca de software de visión artificial y aprendizaje automático de código abierto. Su creación se dio con el fin de suministrar infraestructura común para apps de visión por computadora y para darle celeridad al uso de la perspectiva de la maquina en los productos comerciales [11].

Este sistema tiene una ventaja en cuanto a eficiencia computacional y con un amplio enfoqueen apps en tiempo real. Está escrito en C y C++ optimizados, y su forma de ejecutarse podría darse en GNU/Linux, Windows, Android, iOS y Mac OS X. De la misma forma, es relevante notar que hay un activo de desarrollo en las interfaces de Python, Ruby, Matlab, y otros lenguajes. OpenCV. Según Bradski & Kaehler [12] este sistema tiene una estructura quecomprende cinco componentes, los cuales son:

- ✓ El CV, comprende algoritmos que son capaces de procesar imágenes y de visión porordenador en dos niveles, tanto básico, como superior.
- ✓ ML, se constituye como un centro de recursos en donde aprende la máquina, que comprende muchos clasificadores estadísticos y métodos de agrupación.
- ✓ HighGUI, tiene a su favor rutinas y funciones de In and Out, para almacenar y cargarvideos e imágenes.

El aporte de Open CV permite a Nao, tener la opción de poder utilizar una app para observar y así reconocer facialmente a las personas. Esta herramienta se considera fundamental dentrodel proceso de reconocimiento facial, ya que a través de esta se puede determinar la simetría de los rostros.

- ✓ CXCore, su estructura es básica y tiene algoritmos de apoyo XML, y funciones gráficas. Recepta datos de CV, MIL y HighGUI.
- ✓ CvAux, es un grupo de algoritmos experimentales.

VISIÓN ARTIFICIAL. Su definición se podría orientar hacia una parte fundamental de la "Inteligencia Artificial" que, a través del uso de métodos adecuados, ofrece la obtención, procesamiento y análisis de cualquier tipo de datos especiales, que surgen de imágenes digitales" [13]. Asimismo, Merchán [14], afirma que la visión artificial tiene como finalidad los siguientes objetivos:

- ✓ Identificar objetos estáticos o en movimiento.
- ✓ Ubicación exacta del objeto.
- ✓ Definir de forma física un objeto.
- ✓ Identificar la forma geométrica o biométrica de objetos o personas.
- ✓ Medir la distancia en la que se encuentra un objeto.

RECONOCIMIENTO FACIAL. "Es un proceso que dada una o varias imágenes de una cara desconocida, selecciona entre las caras registradas en su base de datos, aquella con un mayor grado de similitud o parecido, devolviéndose la identidad de ésta" [15]. Por su parte, Castaño & Alonso [16], afirmaron que, dentro de un sistema de reconocimiento facial, la cámara es parte fundamental, debido a que esta es la que toma las fotos del usuario. Sus principales características son:

- ✓ Atrapa imágenes en una calidad alta.
- ✓ Comprende una amplia calidad de información.
- ✓ Posee un enfoque fijo de 8 megapíxeles.
- ✓ Presenta un alto grado de compatibilidad con 1080p30, 720p60 y VGA90.
- ✓ Utiliza un sensor de imagen CMOS Sony IMX219PQ.
- ✓ Tiene un cable plano de quince contactos.

HAAR CASCADES. Esta técnica tiene como función la concatenación de varios clasificadoresdébiles, cada uno con una función diferente de análisis de objetos o personas. Presentan una limitación en cuanto a la probabilidad de dar falso positivo, sin embargo, al estar combinados los resultados, en conjunto, por el contrario, son muy potentes [17]. Principalmente el algoritmo precisa de muchas imágenes positivas, por ejemplo, de rostros y otras negativas, pueden ser de objetos, de tal forma que el clasificador vaya almacenando datos.

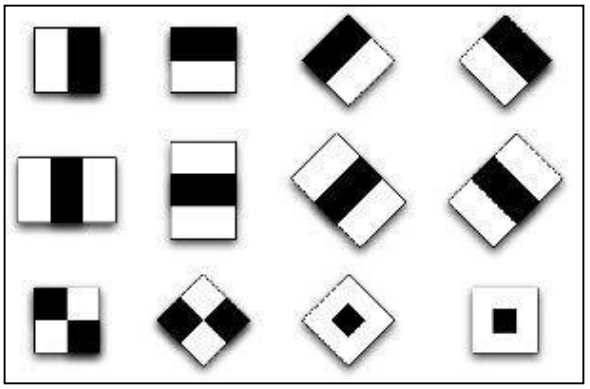


Fig. 3. Clasificador Haar Cascade [17]

Python se podría constituir como lenguaje de programación interpretado, que funciona con el fin de ser legible por cualquiera que pueda tener conocimientos menores de programación. Asimismo, tiene una secuencia de propiedades que lo hacen bastante especial y que, sin lugara duda, le atribuyen ventajas y permanecen en la base de su amplia utilidad. Este lenguaje es de enorme trascendencia en el planeta para las organizaciones, debido a que, por medio de este, se tienen la posibilidad de edificar aplicaciones web, examinar información, automatizar operaciones y generar aplicaciones empresariales fiables y escalables [18].

LBPH FACE RECOGNITION ALGORITHM. El LBPH es un algoritmo de reconocimiento que puede reconocer rostros humanos, a su vez, comprende la capacidad de identificar caras enuna imagen y luego vincularlas a una persona en particular. Este reconocimiento facial se podría lograr, colocando en práctica la técnica, haciendo pruebas con imágenes [19]. Generalmente este algoritmo LBPH hace uso de cuatro parámetros importantes que son:

- ***Radio:*** la distancia del patrón binario local circular desde el píxel central hasta su circunferencia y, por lo general, toma el valor de 1.
- ***Vecinos:*** el número de puntos de datos dentro de un patrón binario local circular. Por logeneral, el valor de 8.
- ***Cuadrícula X:*** El número de celdas en el plano horizontal, suele ser un valor de 8.
- ***Cuadrícula Y:*** El número de celdas en el plano vertical, suele ser un valor de 8.

Con la utilización de estos parámetros, se deben seguir ciertos pasos que son importantes dentro de su operación. Estos pasos según Yegua [19] son:

- **Primer paso.** Se ha creado una imagen intermedia para representar mejor la imagen originala través de un concepto de ventana deslizante, teniendo en cuenta dos parámetros: el *neighbor*y el *radius*. Los nuevos valores se crean en forma binaria comparando los 8 neighbor valores con el valor de umbral.

- **Segundo paso.** Para cada *neighbor* valor superior al valor de umbral, el valor se establece en 1 y 0 para cada *neighbor* valor inferior al valor de umbral. Esto forma una matriz de números binarios excluyendo el umbral. Se crea un valor central de la matriz mediante la conversión del número binario a un valor decimal que corresponde a los píxeles de la imagen original. Parauna mejor representación de las características de la imagen original.

METODOLOGÍA

La metodología de esta investigación partió del desarrollo de un programa en el software Pycharm, mediante el lenguaje de programación Python y usando las librerías de Open CV paralograr el principal objetivo del reconocimiento artificial y una comunicación estable por parte del Robot Nao que se encuentra disponible en la Universidad Politécnica Salesiana.

Como parte del proceso se va a importar las librerías a utilizar como son: 'Pynaoqi', la librería de OpenCV ('cv2'), el algoritmo Haar Cascade y muchas otras más que se vayan a necesitar. Luego se procederá a crear una carpeta donde se guardará fotos de los rostros de las personas que estarán en la base de datos del Robot Nao en diferentes ángulos para que el Robot pueda reconocer y procesar las imágenes de los rostros con ayuda de las librerías mencionadas anteriormente; y por último se desarrollara el algoritmo en el lenguaje de programación 'Python'.

Para poder apreciar que se logren todos los objetivos establecidos, se harán constantemente pruebas del algoritmo realizado en el Robot humanoide Nao. Los resultados de Python se observaron de la siguiente manera:

CONEXIÓN CON EL ROBOT NAO

Para realizar la conexión con el Robot Nao, en primer lugar, se coloca la IP del Robot, el puerto de comunicación y se procede el enlace con la cámara del NAO.

```
ip_addr = "192.168.0.100"
port_num = 9559

videoDevice = ALProxy('ALVideoDevice', ip_addr, port_num)
```

Fig. 4. Conexión del Robot Nao

Se inicia el comando para la voz del NAO y se selecciona el idioma de la voz en español. Por consiguiente, se ingresa a la cámara del Robot y se escoge la cámara superior del NAO. Para la resolución de la cámara se escoge 320x240 y se crea un cuadro con dimensiones de ancho 320 y alto 240 para la imagen que se visualizará en la cámara.

```
tts = ALProxy("ALTextToSpeech", ip_addr, 9559)
tts.setLanguage("Spanish")


AL_kTopCamera = 0
AL_kQVGA =  1
AL_kBGRColorSpace = 13

captureDevice = videoDevice.subscribeCamera("test", AL_kTopCamera, AL_kQVGA, AL_kBGRColorSpace, 10)


width = 320
height = 240
image = np.zeros((height, width, 3), np.uint8)
```

Fig. 5. Configuración del Nao

Se establece la ruta donde están ubicadas las carpetas con los rostros y se escoge la librería para el reconocimiento facial como se muestra en la Fig. 6.

```
dataPath = 'C:/Users/DOUGLAS/Desktop/pythonProject/DATA'
imagePaths = os.listdir(dataPath)
print('imagePaths=',imagePaths)

faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.xml')
```

Fig. 6. Librería Haar Cascade

Se procede a leer el modelo que se creó con el entrenamiento.

```
face_recognizer = cv2.face.LBPHFaceRecognizer_create()
face_recognizer.read('modeloLBPHFace.xml')

g = 0
count =0
lista = []
i = 0
for i in range(len(imagePaths)):
    lista.append(i)
    i += 1

while True:
    #"""
```

Fig. 7. Lectura del modelo de entrenamiento

CONEXIÓN CON CÁMARA DEL NAO

Para establecer la comunicación con la cámara del Nao, se debe utilizar la función 'VideoDevice.getImageRemote' para establecer la conexión. Si en caso no aparezca la cámara, se colocará un escrito que dirá 'cannot capture' indicando que no se ha habilitado la cámara del Robot.

```
camara = videoDevice.getImageRemote(captureDevice);

if camara == None:
    print 'cannot capture.'
    break
if camara[6] == None:
    print 'no image data string.'
    break
else:
    #Valor a matriz
    values = map(ord, list(camara[6]))
    i = 0
    for y in range(0, height):
        for x in range(0, width):
            image.itemset((y, x, 0), values[i + 0])
            image.itemset((y, x, 1), values[i + 1])
            image.itemset((y, x, 2), values[i + 2])
            i += 3
```

Fig. 8. Conexión Cámara del Nao

Se transforma la imagen a escala de grises y se re-escala la imagen a una resolución de 150x150 pixeles. Al detectar los rostros, se escriben los valores de los resultados del reconocimiento; si el rostro coincide con las imágenes guardadas previamente el valor estará dentro del rango, el Robot NAO escribirá el nombre de la persona que detectó guardada en su base de datos en un rectángulo verde alrededor del rostro y dirá el nombre de la persona.

```
gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
auxFrame = gray.copy()

faces = faceClassif.detectMultiScale(gray,1.3,5)

for(x,y,w,h) in faces:
    rostro = auxFrame[y:y+h,x:x+w]
    rostro = cv2.resize(rostro,(150,150),interpolation= cv2.INTER_CUBIC)
    result = face_recognizer.predict(rostro)

    cv2.putText(image,'{}'.format(result),(x,y-5),1,1.3,(255,255,0),1,cv2.LINE_AA)
```

Fig. 9. Código para Detección de Rostros

Si el rostro no coincide con los que están guardados en su base de datos, aparecerá un rectángulo en rojo y se escribirá desconocido.

```
if result[1] < 78 and g==0:
    cv2.putText(image,'{}'.format(imagePaths[result[0]]),(x,y-25),2,1.1,(0,255,0),1,cv2.LINE_AA)
    cv2.rectangle(image,(x,y),(x+w,y+h),(0,255,0),2)
    nombre = "Hola {}".format(imagePaths[result[0]])
    if result[0] in lista:
        print(nombre)
        tts.say(nombre)
        saludo()
        lista.remove(result[0])

if result[1] > 79 and g==0:
    cv2.putText(image, 'Desconocido'.format(imagePaths[0]), (x,y-20), 2, 0.8, (0, 0, 255), 1, cv2.LINE_AA)
    cv2.rectangle(image, (x, y), (x + w, y + h), (0, 0, 255), 2)
```

Fig. 10. Código para Rostros no guardados en base de datos

GUARDADO DE ROSTROS

Para el guardado de rostros, primero se ingresa el nombre de la persona, la cual se guardará el rostro. Luego se presionará la letra 'g' y tomará las capturas de los rostros. Este proceso hará que se guarden 300 fotos del rostro de la persona ingresada en distintos ángulos.

```
k = cv2.waitKey(1)
if k == 103 or g==1:
    if g==0:
        personName = raw_input("Ingrese Nombre: ")
        personPath = dataPath + '/' + personName

        if not os.path.exists(personPath):
            print('CapertaCreada:', personPath)
            os.makedirs(personPath)
    g = 1
```

Fig. 11. Creación de base de datos

```
    cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)
    rostro = auxFrame[y:y + h, x:x + w]
    rostro = cv2.resize(rostro, (150, 150), interpolation=cv2.INTER_CUBIC)
    cv2.imwrite(personPath + '/rostro_{}.jpg'.format(count), rostro)
    count = count + 1
    print(count)

    if k == 27 or count >= 50:
        print('Rostro capturado')
        g = 0
```

Fig. 12. Guardado de Rostros

Si se presiona la letra 'e' realizará la función de entrenamiento y, por consiguiente, abrirá la cámara del NAO, la cual hará el reconocimiento de la persona que se encuentre en frente del Robot. Presionar la tecla 'Esc' ayudará a cancelar el entrenamiento o cerrar la cámara del Robot.

```
    if k == 101:
        Entrenamiento(dataPath)
        imagePaths = os.listdir(dataPath)
        print('imagePaths=', imagePaths)
        # Leyendo el modelo
        faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
        face_recognizer = cv2.face.LBPHFaceRecognizer_create()
        face_recognizer.read('modeloLBPHFace.xml')
        print("Modelo Leido")
        lista = []
        i = 0
        for i in range(len(imagePaths)):
            lista.append(i)
            i += 1
    if k == 27:
        break
```

Fig. 13. Comandos para el entrenamiento

```
    imshow = cv2.imshow('Reconocimiento Facial', image)
    k = cv2.waitKey(1)
    if k == 27:
        break
    if k == 101:
        Entrenamiento(dataPath)
        imagePaths = os.listdir(dataPath)
        print('imagePaths=', imagePaths)
        # Leyendo el modelo
        faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
        face_recognizer = cv2.face.LBPHFaceRecognizer_create()
        face_recognizer.read('modeloLBPHFace.xml')
        print("Modelo Leido")
        lista = []
        i = 0
        for i in range(len(imagePaths)):
            lista.append(i)
            i += 1
videoDevice.unsubscribe(captureDevice)
cv2.destroyAllWindows()
```

Fig. 14. Uso de la librería LBPHFaceRecognizer

ENTRENAMIENTO DEL ALGORITMO

Como parte del procedimiento, se establecerá la Función para el entretenimiento. La primera función 'peopleList' consiste en hacer una lista con los nombres de los rostros guardados en la carpeta DATA. Posteriormente se procede a imprimir la lista.

```
def Entrenamiento(dataPath):
    peopleList = os.listdir(dataPath)
    print('Lista de personas: ', peopleList)
```

Fig. 15. Creación de lista de Rostros guardados

Se procede de igual manera a declarar las variables que se necesitan:

```
labels = []
facesData = []
label = 0
```

Fig. 16. Declaración de variables

Con esta función se hará un recorrido por las fotos de cada carpeta:

```
for nameDir in peopleList:
    personPath = dataPath + '/' + nameDir
    print('Leyendo las imagenes')

    for fileName in os.listdir(personPath):
        print('Rostros: ', nameDir + '/' + fileName)
        labels.append(label)
        facesData.append(cv2.imread(personPath +'/'+fileName,0))
        image = cv2.imread(personPath +'/'+fileName,0)
        cv2.imshow('Captura',image)
        cv2.waitKey(10)
    label = label+1
```

Fig. 17. Lectura de las imágenes guardadas

Se crea el código para el método LBPH Face Recognizer, para el entrenamiento y se realiza el código para el reconocimiento facial. Por consiguiente, se guarda el modelo que se crea con el entrenamiento como se muestra en la Fig. 18.

```
face_recognizer = cv2.face.LBPHFaceRecognizer_create()

print('Entrenando...')
face_recognizer.train(facesData, np.array(labels))

face_recognizer.write('modeloLBPHFace.xml')

print('Modelo almacenado')
cv2.destroyWindow('Captura')
```

Fig. 18. Entrenamiento del modelo

MOVIMIENTO DE SALUDO DE NAO

Para realizar el movimiento del saludo por parte del Robot, se utiliza el software Choreographe, que ayuda a crear movimientos y reacciones de NAO. Se usa la función 'Timeline' para configurar cada movimiento de la mano y guardarlos en la línea de tiempo para lograr el saludo.

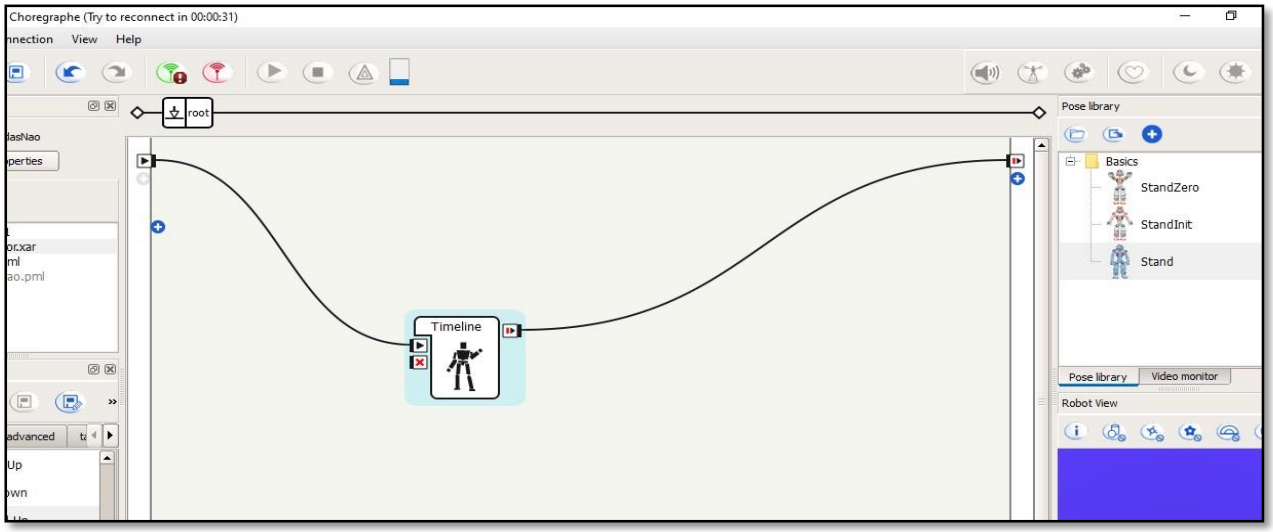
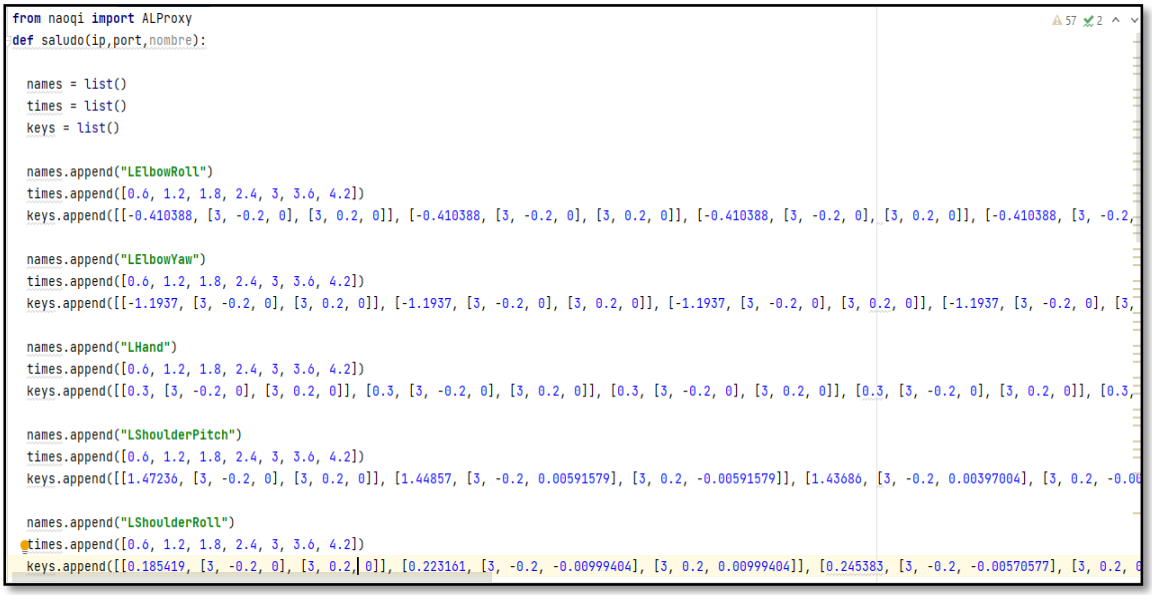


Fig. 19. Creación de movimiento de saludo en Choreographe

Posteriormente, ya guardado cada movimiento de la mano para que el Robot realice el saludo,se exporta el proyecto en lenguaje de programación Python, para poder integrar el código obtenido en nuestro código general, haciendo que NAO al momento de reconocer a la persona que está registrada en su base de datos, realice la reacción de saludo.

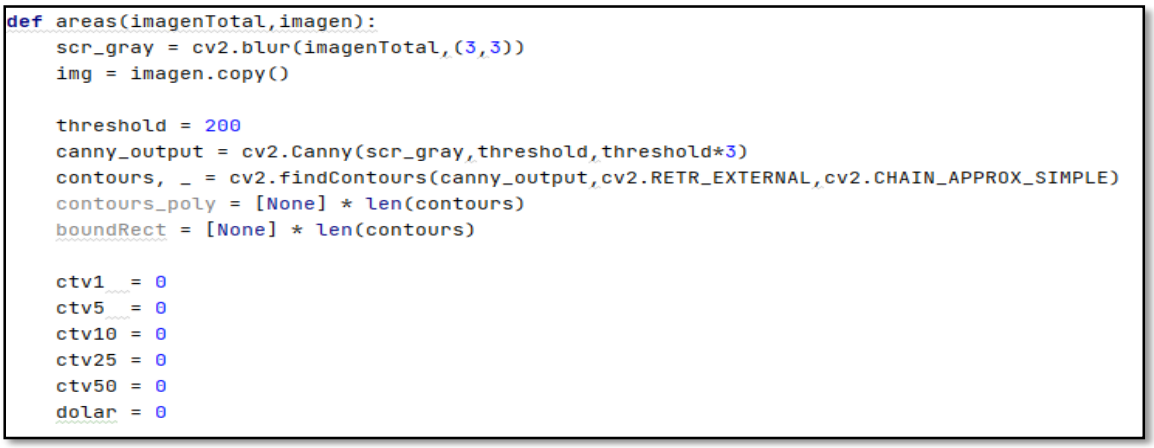


```python
from naoqi import ALProxy
def saludo(ip,port,nombre):

  names = list()
  times = list()
  keys = list()

  names.append("LElbowRoll")
  times.append([0.6, 1.2, 1.8, 2.4, 3, 3.6, 4.2])
  keys.append([[-0.410388, [3, -0.2, 0], [3, 0.2, 0]], [-0.410388, [3, -0.2, 0], [3, 0.2, 0]], [-0.410388, [3, -0.2, 0], [3, 0.2, 0]], [-0.410388, [3, -0.2,

  names.append("LElbowYaw")
  times.append([0.6, 1.2, 1.8, 2.4, 3, 3.6, 4.2])
  keys.append([[-1.1937, [3, -0.2, 0], [3, 0.2, 0]], [-1.1937, [3, -0.2, 0], [3, 0.2, 0]], [-1.1937, [3, -0.2, 0], [3, 0.2, 0]], [-1.1937, [3, -0.2, 0], [3,

  names.append("LHand")
  times.append([0.6, 1.2, 1.8, 2.4, 3, 3.6, 4.2])
  keys.append([[0.3, [3, -0.2, 0], [3, 0.2, 0]], [0.3, [3, -0.2, 0], [3, 0.2, 0]], [0.3, [3, -0.2, 0], [3, 0.2, 0]], [0.3, [3, -0.2, 0], [3, 0.2, 0]], [0.3,

  names.append("LShoulderPitch")
  times.append([0.6, 1.2, 1.8, 2.4, 3, 3.6, 4.2])
  keys.append([[1.47236, [3, -0.2, 0], [3, 0.2, 0]], [1.44857, [3, -0.2, 0.00591579], [3, 0.2, -0.00591579]], [1.43686, [3, -0.2, 0.00397004], [3, 0.2, -0.0

  names.append("LShoulderRoll")
  times.append([0.6, 1.2, 1.8, 2.4, 3, 3.6, 4.2])
  keys.append([[0.185419, [3, -0.2, 0], [3, 0.2, 0]], [0.223161, [3, -0.2, -0.00999404], [3, 0.2, 0.00999404]], [0.245383, [3, -0.2, -0.00570577], [3, 0.2, 0
```

Fig.20. Exportación de movimiento de saludo en Python

RECONOCIMIENTO DE MONEDAS

Para proceder con el reconocimiento de monedas por parte del Robot, se definen las variables a utilizar. Luego, se aplican los algoritmos de detección de bordes ‘Canny’ y ‘FindContours’ para hallar los contornos de las monedas.



```python
def areas(imagenTotal,imagen):
    scr_gray = cv2.blur(imagenTotal,(3,3))
    img = imagen.copy()

    threshold = 200
    canny_output = cv2.Canny(scr_gray,threshold,threshold*3)
    contours, _ = cv2.findContours(canny_output,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
    contours_poly = [None] * len(contours)
    boundRect = [None] * len(contours)

    ctv1  = 0
    ctv5  = 0
    ctv10 = 0
    ctv25 = 0
    ctv50 = 0
    dolar = 0
```

Fig. 21. Definición de variables

```
mu = [None] * len(contours)
for i in range(len(contours)):
    mu[i] = cv2.moments(contours[i])

mc = [None] * len(contours)
for i in range(len(contours)):
    mc[i] = (mu[i]['m10'] / (mu[i]['m00'] + 1e-5), mu[i]['m01'] / (mu[i]['m00'] + 1e-5))
drawing = np.zeros((canny_output.shape[0], canny_output.shape[1],3),dtype=np.uint8)

for i in range(len(contours)):
    color = (random.randint(0,256),random.randint(0,256),random.randint(0,256))
    cv2.circle(drawing,(int(mc[i][0]),int(mc[i][1])),4,color,-1)
    color2 = 255,255,255
    ren = (round(mc[i][0]),round(mc[i][1]))
    area = cv2.contourArea(contours[i])
    print(area)
    cv2.drawContours(drawing,contours,i,color2,2)
```

Fig. 22. Detección de Contornos de Monedas

Se define el área de las circunferencias de cada moneda, las cuales se van a colocar en una superficie para poder detectarlas con el NAO. Al ser detectadas múltiples monedas, se sumarán entre sí y el Robot dirá el valor total de la equivalencia de las monedas.

```
        if 130 < area <165:
            ctv10 = ctv10 +1
            time.sleep(0.0125)
        if 170 < area <225:
            ctv5 = ctv5 +1
            time.sleep(0.0125)
        if 255 < area <320:
            dolar = dolar +1
            time.sleep(0.0125)
        if 325 < area <405:
            ctv50 = ctv50 +1
            time.sleep(0.0125)

    total = (ctv1*0.01)+(ctv10*0.1)+(ctv5*0.05)+(ctv25*0.25)+(ctv50*0.5)+(dolar*1)
    total = round(total,2)

    cv2.imshow('Contador de monedas',img)
    return total

total1 = 50.0
while True:
```

Fig. 23. Definir área de Monedas

Por último, se convierte la entrada en un arreglo, las cuales se colocan en las variables asignadas. Se aplica la función 'cv2.filter2D' para cambiar el valor de intensidad de pixeles de una imagen para mejorar o eliminar ciertas características y así crear otra imagen. Luego se aplican intervalos a los contornos de la imagen con la función 'np.clip' como se puede apreciar en la Fig. 25.

```
im = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)

a = np.asarray([[1.0, 0.0, -1.0],
                [1.0, 0.0, -1.0],
                [1.0, 0.0, -1.0]])
b = np.asarray([[-1.0, 0.0, 1.0],
                 [-1.0, 0.0, 1.0],
                 [-1.0, 0.0, 1.0]])
c = np.asarray([[1.0, 1.0, 1.0],
                 [0.0, 0.0, 0.0],
                 [-1.0, -1.0, -1.0]])
d = np.asarray([[-1.0, -1.0, -1.0],
                 [0.0, 0.0, 0.0],
                 [1.0, 1.0, 1.0]])
e = np.asarray([[0, 0, -1, 0, 0],
                 [0, 0, -1, 0, 0],
                 [0, -1, 5, -1, 0],
                 [0, 0, -1, 0, 0],
                 [0, 0, 0, 0, 0]])
```

Fig. 24. Creación de arreglos

```
dst  = cv2.filter2D(im,-1,a); dst = np.clip(dst,0,255)
dst2 = cv2.filter2D(im,-1,b); dst2 = np.clip(dst2,0,255)
dst3 = cv2.filter2D(im,-1,c); dst3 = np.clip(dst3,0,255)
dst4 = cv2.filter2D(im,-1,d); dst4 = np.clip(dst4,0,255)
dst5 = cv2.filter2D(im,-1,e); dst5 = np.clip(dst5,0,255)

imagenTotal = dst | dst2 | dst3 | dst4 | dst5

total2 = areas(imagenTotal,image)
if total2 != total1 and total2 > 0.1:
    tts.say('{} Dolar'.format(total2))
    total1 = total2
```

Fig. 25. Aplicación de Intervalos de Contorno a Imágenes

RESULTADOS

En los resultados de este estudio, se presenta el análisis realizado de la aplicación del Robot Nao, comprobando la efectividad del reconocimiento facial en las personas que se encuentran ingresadas en su base de datos.

ROBOT NAO Y RECONOCIMIENTO FACIAL

Para las pruebas de análisis de reconocimiento facial, se trabaja de forma conjunta con el rastreo de rostros el cual permite hacer la identificación. Este procedimiento consiste en que el robot se nutra de la información del set de datos, para que posteriormente al hacer el reconocimiento facial, este pueda comparar la información y determinar el reconocimiento de la persona.

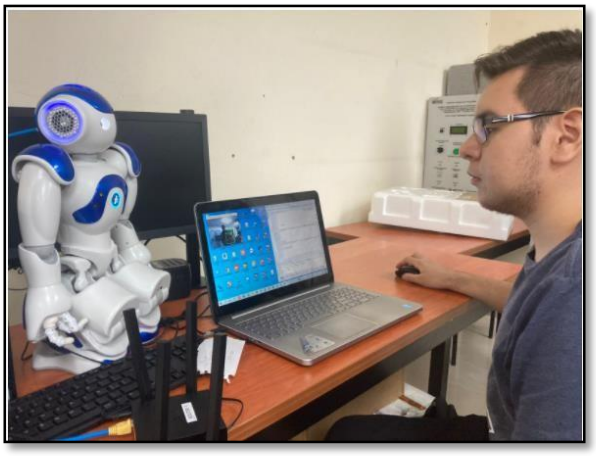


Fig. 26. Reconocimiento Facial Prueba #1

Al momento de realizar el reconocimiento facial, el robot debe estar conectado al set de datos. Se tuvo que utilizar un router para poder incrementar la velocidad de la cámara que va a detectar los rostros de las personas que estarán guardados en su base de datos, ya que al momento de utilizar la red Wifi de la Universidad Politécnica Salesiana, el entrenamiento del algoritmo en la captura de rostros se desarrollaba de manera muy lenta, por lo cual la solución era requerir de un router e interconectar tanto los diferentes dispositivos a utilizar con el Robot.

Fig. 27. Reconocimiento Facial Prueba #2

La posición del robot debe ser de frente con la persona que se va someter al reconocimiento facial, de esta forma va poder realizar una mejor lectura de los rostros.

PRUEBAS DE RECONOCIMIENTO FACIAL

Para este caso, se realiza la prueba de reconocimiento facial con dos personas, con las cuales se demuestra la efectividad del robot a la hora de realizar el reconocimiento facial. Luego, se ingresan más rostros para el reconocimiento y se observa que el NAO puede reconocer más de 3 rostros y los haga un saludo, cumpliendo con el objetivo planteado.

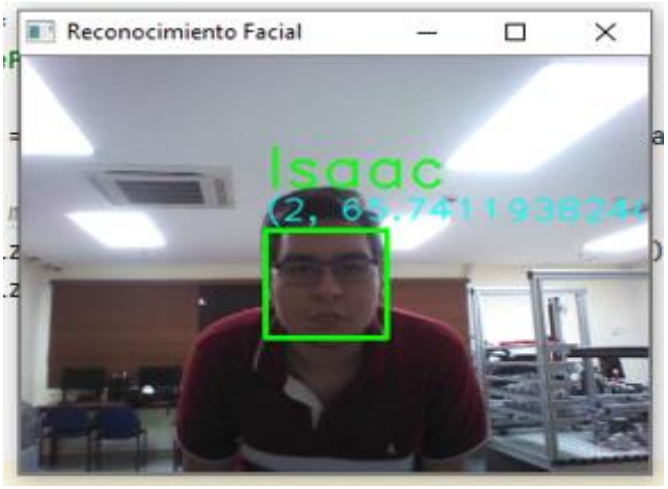


Fig. 28. Resultado de Reconocimiento Facial Prueba #1

Fig. 28. Resultado de Reconocimiento Facial Prueba #2

En este punto se realiza el reconocimiento facial, a través de la cámara del robot, que al mismo tiempo hace la comparativa con el set de datos para determinar la identidad de la persona que está sometida a la prueba de reconocimiento facial.

PRUEBAS DE RECONOCIMIENTO DE MONEDAS

Para hacer el reconocimiento de monedas, se coloca una superficie blanca con las monedas a detectar. Para este proceso, se utiliza la cámara interna inferior del Robot Nao, la cual permitía visualizar el área de la circunferencia de cada moneda en la superficie designada. Al darnos cuenta de que el Robot detecta la moneda en la cámara, se delimita el área de trabajo para colocar las monedas y así no tener problemas en cuanto las monedas estén en un punto ciego en donde Nao no las pueda ver.

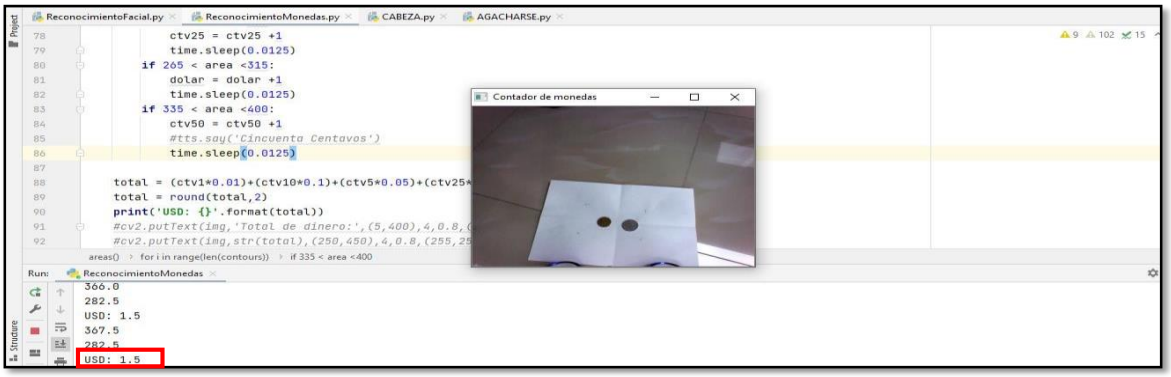


Fig. 29. Resultado de Reconocimiento de Monedas Prueba #1

Se efectúan múltiples pruebas con cada moneda para saber que el robot las detectaba y luego se realizan pruebas con diversas monedas en conjunta para ver si Nao las sumaba e indicaba la equivalencia de cada moneda o su total con monedas en conjunto. Al realizar las pruebas se puede apreciar, que se cumple con éxito la parte del proyecto en cuanto al reconocimiento de las monedas, la cual era parte de los objetivos planteados a realizar.

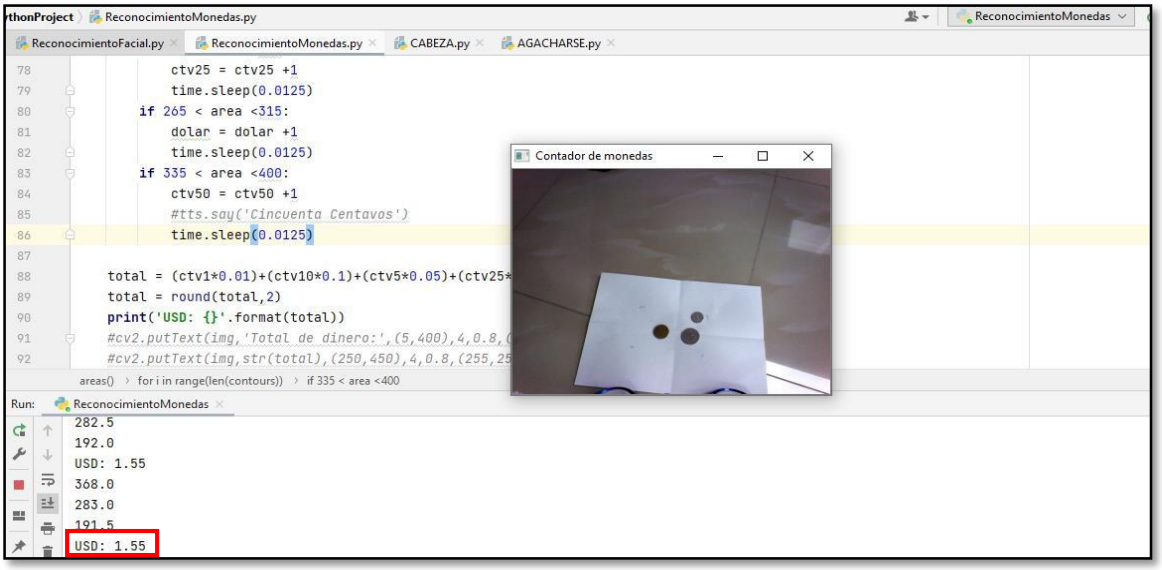


Fig. 30. Resultado de Reconocimiento de Monedas Prueba

Por último, se realizan pruebas con una cámara externa para ver qué tan precisa es la cámara del Nao al momento de detectar todas las monedas y se puede observar que la cámara externa tiene una mejor resolución que permite obtener una mayor rapidez y nitidez al momento de detectar cada moneda por parte del Robot.

Fig. 31. Comparación Cámara Externa Vs. Cámara del Robot Nao

CONCLUSIONES

- Con la investigación realizada, se determinó que el Robot Nao podría tener múltiples funciones, tanto de reconocimiento facial, como reconocimiento de voz y de expresiones. En este caso la opción de reconocimiento facial mostró un resultado óptimo, por lo que se podría concluir que la herramienta podría usarse para esta función en la Universidad.

- El lenguaje de programación Python permitió que el tratamiento de la información para elrobot se optimice y los resultados vayan más orientados a la necesidad presentada. La facilidad de este lenguaje de programación permitió que el funcionamiento de Nao fueraentendible, aun para personas que no son expertos en programación.

- Se creó correctamente una base de datos para el almacenamiento de los rostros de las personas que serán reconocidas por el Robot Nao.

- Se logró obtener el reconocimiento facial a través de la programación en lenguaje Python y OpenCV mediante los algoritmos: 'Haar Cascade' y 'LBPH Face Recognizer', este último algoritmo previamente entrenado arrojará un valor dentro de un rango específico el cual servirá para identificar si el rostro de la persona ha sido guardado en la base de datos.

- El Robot Nao saludará con un leve movimiento de la mano y dirá el nombre de la persona a la cual reconoció.

- Se aplicó correctamente un reconocimiento de monedas, en el cual se detectará los contornos de cada moneda calculando el área y así poder determinar que moneda es la que está viendo el Robot Nao.

- A través de la realización de una prueba con una cámara externa, con la finalidad de probar la efectividad de la cámara de Nao, se colocaron múltiples monedas de diferentes valores, incluyendo una de veinticinco centavos, que almomento de someterlas a prueba, se notó que la cámara del Robot no reconocía la moneda de

veinticinco centavos, ya que la resolución de la cámara propia de Nao, es muy baja y generó confusión entre la moneda de un dólar y veinticinco centavos. Por lo que se concluye que la utilización de una cámara externa con mejor resolución hizo el reconocimiento exacto de todas las monedas, incluyendo la de veinticinco centavos, que era la que presentaba problema.

- Se podría concluir que el uso de otros lenguajes de programación podría orientar la investigación hacia otro resultado distinto, es decir, que se programe una nueva función para el Robot, considerando que comprende una serie de funciones a las cuales se les podría sacar provecho, dentro de estas funciones se destacan como antecedentes principales, la interpretación de expresiones faciales, la tele-operación móvil yprogramación de Nao para jugar fútbol. Cabe resaltar que, dentro de estos antecedentes,pueden surgir diversas funciones adicionales, que podrían verse como objeto de estudio para futuras investigaciones.

REFERENCIAS BIBLIOGRÁFICAS

[1] SoftBank Robotics, «Robotrónica,» 2020. [En línea]. Available: https://aliverobots.com/nao/.

[2] Calvopiña F. & Valladares F., «Interpretacion de expresiones faciales de adultos mayores utilizano la vision artificial del robot humanoide NAO,» febrero 2017. [En línea]. Available: https://bit.ly/3J43qwp.

[3] D. R. P. Abhishek Kumar Kashyap, «Optimization of stability of humanoid robot NAO using ant colony, » 05 Enero 2021. [En línea]. Available: https://bibliotecas.ups.edu.ec:3401/content/pdf/10.1007/s00500-020-05515-1.pdf.

[4] L. Garrido, «Transferencia Tec,» 8 Abril 2018. [En línea]. Available: https://transferencia.tec.mx/2018/04/08/robot-nao-una-terapia-para-ninos-con-trastornos-de- neurodesarrollo/.

[5] Cortés F., Robotica, Control de Robots Manipuladores, Marcombo S.A, 2011.

[6] Molina M., «Límites a la Inteligencia Artificial,» 24 Julio 2018. [En línea]. Available: https://bit.ly/34ey5XK.

[7] Rivas W. & Mazón B., Redes neuronales artificiales aplicadas al reconocimiento de patrones, México
: © Editorial UTMACH, 2018.

[8] Gonzáles A., « ¿Qué es Machine Learning?,» 2019. [En línea]. Available: https://bit.ly/3HqLoTM.

[9] Delgado S., «Aprende Python,» 10 noviembre 2021. [En línea]. Available: https://bit.ly/3ebceT1.

[10] Martelli A., Python in a Nutshell: A Desktop Quick Reference (2da ed.). Estado Unidos: O'Reilly Media, Inc., 2006.

[11] Tungay K., «Construyendo un reconocedor de color en Python,» 2020. [En línea]. Available: https://bit.ly/3mi7qj6.

[12] Bradski G. & Kaehler A., Learning OpenCV, Sebastopol: O'Reilly Media, Inc, 2008.

[13] Folgado E., «Easy Tech: La visión artificial como paradigma del presente y futuro,» 24 octubre 2019. [En línea]. Available: https://bit.ly/3GZAjsf.

[14] Merchán E., «Sistema de Visión Artificial para el Control de Calidad de Piezas Cromadas,» 12 noviembre 2009. [En línea]. Available: https://bit.ly/3pikC9A.

[15] Belén A. Díaz M. & Físicas C., «Reconocimiento Facial Automático mediante Técnicas de Visión Tridimensional Tesis Doctoral Reconocimiento,» 2004. [En línea]. Available: https://bit.ly/3ehoj93.

[16] Castaño D. & Alonso J., «SISTEMA DE RECONOCIMIENTO FACIAL PARA CONTROL DE ACCESO A
VIVIENDAS,» 2019. [En línea]. Available: https://bit.ly/3Ejyl4r.

[17] Jeremías E., «Reconocimiento de objetos a través de la metodología Haar Cascades,» *REVISTAARGENTINA DE INGENIERÍA - AÑO 8 - VOLUMEN 16 ISSN 2314-0925,* pp. 1-7, 2020.

[18] Visus A., « ¿Para qué sirve Python? Razones para utilizar este lenguaje de programación,» Octubre2020. [En línea]. Available: https://bit.ly/34si7cM.

[19] Yegua R., «Comprensión del reconocimiento facial mediante el algoritmo de histograma de patronesbinarios locales (LBPH),» 16 Julio 2021. [En línea]. Available: https://bit.ly/3Ho5iif.

[20] LISA Institute, «LISA Institute,» 07 Abril 2021. [En línea]. Available: https://www.lisainstitute.com/blogs/blog/reconocimiento-facial-como-funciona-quien-utiliza.

[21] Packt Publishing Ltda, «Community Experience Distilled,» de *OPEN CV Computer Vision with Python*,Birmingham, UK, Packt Publishing Ltda, 2013, p. 122.

[22] Alvarez M., «Lenguaje de programación de propósito general, orientado a objetos, que también puede utilizarse para el desarrollo web,» 19 noviembre 2003. [En línea]. Available: https://bit.ly/3eg0sGM.