



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

CONTROL DE UN BRAZO ROBÓTICO PARA RECONOCIMIENTO DE
OBJETOS UTILIZANDO SENSORES DE VISIÓN INTELIGENTE

ESPINOZA PARDO ELISA GABRIELA
INGENIERA DE SISTEMAS

MACHALA
2021



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

CONTROL DE UN BRAZO ROBÓTICO PARA
RECONOCIMIENTO DE OBJETOS UTILIZANDO SENSORES DE
VISIÓN INTELIGENTE

ESPINOZA PARDO ELISA GABRIELA
INGENIERA DE SISTEMAS

MACHALA
2021



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

TRABAJO TITULACIÓN
PROPUESTAS TECNOLÓGICAS

CONTROL DE UN BRAZO ROBÓTICO PARA RECONOCIMIENTO DE OBJETOS
UTILIZANDO SENSORES DE VISIÓN INTELIGENTE

ESPINOZA PARDO ELISA GABRIELA
INGENIERA DE SISTEMAS

RIVAS ASANZA WILMER BRAULIO

MACHALA, 27 DE ABRIL DE 2021

MACHALA
2021

elisa espinoza

INFORME DE ORIGINALIDAD

8%

INDICE DE SIMILITUD

7%

FUENTES DE INTERNET

0%

PUBLICACIONES

3%

TRABAJOS DEL
ESTUDIANTE

ENCONTRAR COINCIDENCIAS CON TODAS LAS FUENTES (SOLO SE IMPRIMIRÁ LA FUENTE SELECCIONADA)

1%

★ Submitted to UNIV DE LAS AMERICAS

Trabajo del estudiante

Excluir citas

Activo

Excluir coincidencias < 15 words

Excluir bibliografía

Activo

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

La que suscribe, ESPINOZA PARDO ELISA GABRIELA, en calidad de autora del siguiente trabajo escrito titulado CONTROL DE UN BRAZO ROBÓTICO PARA RECONOCIMIENTO DE OBJETOS UTILIZANDO SENSORES DE VISIÓN INTELIGENTE, otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

La autora declara que el contenido que se publicará es de carácter académico y se enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

La autora como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 27 de abril de 2021


ESPINOZA PARDO ELISA GABRIELA
0705048106

DEDICATORIA

Dedico este trabajo de manera especial a mi hijo, que es el pilar fundamental y soporte de mi vida, a mis padres y hermano que, a través de sus consejos y valores me brindan su apoyo incondicional en cada una de las metas que me he propuesto durante mi vida personal como de estudiante.

Elisa Gabriela Espinoza Pardo

AGRADECIMIENTO

En primer lugar, agradezco a Dios por permitirme continuar con mis estudios académicos y darme la fortaleza de culminarlos.

A mis padres quienes en el transcurso de la carrera me han apoyado mediante su cuidado y comprensión, pero sobre todo por el apoyo incondicional brindado.

Y por último a todos los docentes que fueron parte de mi vida estudiantil y de manera especial a mi tutor el ingeniero Wilmer Rivas quienes, gracias a sus enseñanzas impartidas, me han permitido adquirir los conocimientos indispensables para una vida profesional.

Elisa Gabriela Espinoza Pardo

RESUMEN

La evolución tecnológica ha crecido exponencialmente modificando y mejorando la calidad de vida de las personas y por ende las diversas actividades que rodean su entorno, dentro de todos estos procesos tecnológicos que pretenden mejorar la capacidad de dar soluciones a posibles problemas y con ello mejorar la toma de decisiones en un entorno tecnificado podemos observar que la tecnología contribuye grandemente en el ámbito educativo puesto que permite integrar ciencia y tecnología mediante procesos que van desde la planificación, evaluación hasta el seguimiento de aportes de gran importancia que ayuden a mejorar la productividad y eficiencia. De ahí que nuestro enfoque se dirija al área de la informática específicamente en lo relacionado a visión inteligente al ser una rama que encuentra estrechamente ligada a nuestra línea profesional.

Como sabemos el ser humano tiene la habilidad natural de reconocer las diversas formas de objetos, cuando se observa se puede detectar sin mucho esfuerzo sus colores, características y diferenciarlas unas de otras. Al contrario de las ciencias computacionales que utiliza diversos mecanismos de la inteligencia artificial y con ello obtener información pertinente que les permita diferenciarlas, sin embargo, llegar a esta detección puede implicar una diversidad de procesos, métodos y técnicas.

La inteligencia artificial permite emular determinadas acciones que son propias del ser humano, por esta razón en la actualidad es un área ampliamente investigada y en pleno auge debido a sus diversas ventajas y utilidades dentro de nuestro entorno. De ahí que este campo de estudio este activo debido a los grandes retos que conlleva: la detección de objetos en distintas condiciones, con distintos tamaños y perspectivas, etc. Puesto que permite la automatización de procesos, así como un mejor control y precisión en actividades predeterminadas.

El presente trabajo describe el proceso de detección de objetos definiendo las siguientes clases: círculo, cuadrado, triángulo a través de un script en Python, se seleccionó este lenguaje de programación porque se adapta a los requerimientos y necesidades para la ejecución de nuestro trabajo puesto que es ideal para el

trabajo con grandes volúmenes de datos y para nuestro caso trabajaremos con imágenes de objetos.

Se debe mencionar que la detección de objetos se hará en tiempo real puesto que los datos recopilados son durante la fase de crecimiento del algoritmo facilitando información que permita detectar puntos de mejora para tomar mejores decisiones ante problemas que se puedan suscitar y con ello mejorar la eficiencia del script.

Se usará un brazo robótico diseñado con funciones específicas y básicas como movimientos rotacionales y de traslación que permitan emular al de un ser humano. También se formará un script en Arduino que dará movimiento al brazo robótico permitiendo controlar sus acciones durante el proceso de reconocimiento de los objetos. Este software es de código abierto y multiplataforma lo que nos facilita para lograr nuestros objetivos.

Para la obtención de resultados, se estableció el uso de métricas de evaluación, como es el caso de una matriz de confusión puesto que en el campo de la inteligencia artificial esta herramienta permite visualizar el desempeño del algoritmo desarrollado y corroborar si los procedimientos realizados son los apropiados.

Palabras clave: Inteligencia Artificial, Reconocimiento de objetos, Python, Brazo robótico.

ABSTRACT

Technological evolution has grown exponentially modifying and improving the quality of life of people and therefore the various activities that surround their environment, within all these technological processes that seek to improve the ability to provide solutions to possible problems and thereby improve the decision. In making decisions in a technified environment, we can observe that technology contributes greatly in the educational field since it allows the integration of science and technology through processes that range from planning, evaluation to the monitoring of contributions of great importance that help to improve productivity and efficiency. Hence, our focus is directed to the area of computing specifically in relation to intelligent vision as it is a branch that is closely linked to our professional line.

As we know, human beings have the natural ability to recognize the various forms of objects, when observed, their colors and characteristics can be detected without much effort and differentiated from each other. Contrary to computer science that uses various mechanisms of artificial intelligence and thereby obtain relevant information that allows them to differentiate them, however, reaching this detection may involve a diversity of processes, methods and techniques.

Artificial intelligence allows us to emulate certain actions that are typical of the human being, for this reason it is currently a widely investigated area and in full swing due to its various advantages and utilities within our environment. Hence, this field of study is active due to the great challenges it entails: the detection of objects in different conditions, with different sizes and perspectives, etc. Since it allows the automation of processes, as well as better control and precision in predetermined activities.

This work describes the object detection process defining the following classes: circle, square, triangle through a Python script, this programming language was selected because it adapts to the requirements and needs for the execution of our work since It is ideal for working with large volumes of data and in our case we will work with images of objects.

It should be mentioned that the detection of objects will be done in real time since the data collected is during the growth phase of the algorithm, providing information that allows detecting points of improvement to make better decisions in the face of problems that may arise and thereby improve efficiency. of the script.

A robotic arm designed with specific and basic functions such as rotational and translational movements will be used to emulate that of a human being. An Arduino script will also be formed that will give movement to the robotic arm, allowing it to control its actions during the object recognition process. This software is open source and multiplatform which makes it easier for us to achieve our goals.

To obtain results, the use of evaluation metrics was established, such as a confusion matrix, since in the field of artificial intelligence this tool allows visualizing the performance of the developed algorithm and corroborating whether the procedures performed are the same. appropriate.

Keywords: Artificial Intelligence, Object Recognition, Python, Robotic Arm

Tabla de contenido

| | |
|--|----|
| DEDICATORIA | 1 |
| AGRADECIMIENTO | 2 |
| RESUMEN | 3 |
| ABSTRACT | 5 |
| INTRODUCCIÓN | 10 |
| 1. DIAGNÓSTICO DE NECESIDADES Y REQUERIMIENTOS | 12 |
| 1.1. Ámbito de Aplicación: descripción del contexto y hechos de interés 12 | |
| 1.2. Establecimiento de requerimientos | 13 |
| 1.3. Justificación de requerimiento a satisfacer | 13 |
| 2. DESARROLLO DEL PROTOTIPO | 14 |
| 2.1. Definición del prototipo tecnológico | 14 |
| 2.2. Fundamentación teórica del prototipo. | 15 |
| 2.2.1. Entorno de Trabajo | 15 |
| 2.2.1.1. Software | 15 |
| 2.2.1.1.1. Diseño Gráfico 3D | 15 |
| 2.2.1.1.1.1. OnShape | 15 |
| 2.2.1.1.2. TinkerCAD | 15 |
| 2.2.1.1.3. OpenCv | 16 |
| 2.2.1.1.4. Python | 16 |
| 2.2.1.1.4.1. Numpy | 16 |
| 2.2.1.2. Capacidad Computacional | 16 |
| 2.2.1.2.1. CPU | 17 |
| 2.2.1.2.2. GPU | 17 |
| 2.2.1.2.3. Cámara Web | 18 |

| | | |
|------------|--|----|
| 2.2.1.2.4. | Brazo Robótico | 18 |
| 2.2.1.2.5. | Impresora 3D..... | 18 |
| 2.2.1.2.6. | Servomotores..... | 18 |
| 2.2.1.2.7. | Arduino..... | 19 |
| 2.2.2. | Técnicas de Tratamiento de Imágenes..... | 19 |
| 2.2.2.1. | Detección de Objetos | 19 |
| 2.2.3. | Algoritmos de detección de objetos..... | 20 |
| 2.2.3.1. | Sublime text..... | 20 |
| 2.3. | Metodología | 20 |
| 2.4. | Objetivos del Prototipo | 21 |
| 2.4.1. | Objetivo General..... | 21 |
| 2.4.2. | Objetivo Especifico | 21 |
| 2.5. | Diseño del prototipo..... | 22 |
| 2.5.1. | Software | 22 |
| 2.5.2. | Requisitos Hardware..... | 22 |
| 2.5.3. | Obtención del modelo para el reconocimiento de Objetos | 23 |
| 2.6. | Ejecución y/o ensamblaje del prototipo | 23 |
| 2.6.1. | Obtener el hardware Requerido | 23 |
| 2.6.2. | Crear los Scripts necesarios para la detección de Objetos | 24 |
| 2.6.2.1. | Scripts para la detección de Objetos | 24 |
| 2.6.2.2. | Scripts para la selección del objeto requerido para la tarjeta Arduino | 24 |
| 3. | EVALUACIÓN DEL PROTOTIPO..... | 25 |
| 3.1. | Plan de evaluación | 25 |
| 3.1.1. | Métricas de Clasificación..... | 25 |
| 3.2. | Resultado de evaluación | 27 |
| 3.2.1. | Resultado de la Prueba de Entrenamiento y Validación..... | 27 |

| | |
|--|----|
| 3.3. Conclusiones | 29 |
| 3.4. Recomendaciones | 30 |
| 4. BIBLIOGRAFÍA | 31 |
| 5. ANEXOS | 36 |
| 5.1. Anexo I. Código fuente archivo “figurasGeometricas.py” | 36 |
| 5.2. Anexo I. Código fuente archivo “AplicaciónCV.ino” | 38 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura 1: Arquitectura del Prototipo | 14 |
| Figura 2: Mapa Mental de la Fundamentación Teórica del Prototipo | 15 |
| Figura 3: Estructura de un CPU | 17 |
| Figura 4: Arduino | 19 |
| Figura 5: Metodología | 21 |
| Figura 6: Flujo de Desarrollo del Sistema | 23 |
| Figura 7: Hardware Requerido | 23 |
| Figura 8: Reconocimiento y etiquetado del objeto (Triangulo) | 27 |
| Figura 9: Reconocimiento y etiquetado del objeto (cuadrado) | 28 |
| Figura 10: Reconocimiento y etiquetado del objeto (circulo)..... | 28 |

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 1: Software Requerido..... | 22 |
| Tabla 2: Hardware Requerido | 22 |
| Tabla 3: Matriz de Confusión..... | 25 |
| Tabla 4: Información de la Prueba 1 | 27 |
| Tabla 5:Matriz de Confusión Prueba 1 | 28 |

INTRODUCCIÓN

Hoy en día podemos ver que es más evidente el uso de la inteligencia artificial en nuestra vida, sin embargo, al ser un área en pleno auge su utilización aún está en crecimiento, la implementación de la inteligencia artificial[1] en la educación es fuente de una gran ventaja puesto que ayuda a impartir y generar conocimientos.

Para ello debemos entender que la inteligencia artificial (IA) trata de emular la inteligencia humana[2], a través de máquinas o sistemas programados para ejercitar acciones humanas, como usar información, razonar, corregir, validar, entre otros, se está aplicando en muchas áreas: educativa, económica, empresarial, etc. Con el propósito de mejorar sus procesos automatizándolos.

La detección de objetos en tiempo real tiene diversos usos y beneficios como: vigilancia de seguridad, medicina, marketing, entre otros. Es importante considerar que el algoritmo para la detección de objetos debe proporcionar confiabilidad, así como el mostrar precisión al momento de dar resultados. Dentro de la detección de objetos nos encontramos ante grandes desafíos como es el detectar o reconocer los objetos en tiempo real.

Se pretende combinar el uso de un brazo robótico[3] y un algoritmo para el reconocimiento de objetos en tiempo real su finalidad es proporcionar guías para el campo de la educación como apoyo en el área de la inteligencia artificial.

Para la obtención de resultados se requiere utilizar un algoritmo de detección de objetos[4] en tiempo real, se debe considerar que tenga una estructura que se adapte a las necesidades requeridas para el entrenamiento del conjunto de datos adquirido.

El presente documento se encuentra estructurado de la siguiente manera:

Capítulo 1: En este capítulo se describe la utilidad en la implementación del control de un brazo robótico para el reconocimiento de objetos, describiendo el contexto de la problemática, la justificación, los requerimientos del prototipo y argumentación para la solución planteada.

Capítulo 2: se detallan definiciones, fundamentación teórica de la implementación; también se describen los objetivos, diseño, ejecución y pruebas de la aplicación.

Capítulo 3: se establecen los resultados de la implementación, así como también se detallan las conclusiones y recomendaciones según los objetivos descritos y acorde a la solución propuesta.

1. DIAGNÓSTICO DE NECESIDADES Y REQUERIMIENTOS

1.1. Ámbito de Aplicación: descripción del contexto y hechos de interés

Los avances tecnológicos han tenido un progreso exponencial en los últimos años y esto se ha visto reflejado en las diversas áreas de las Ciencias Computacionales, específicamente nos enfocaremos en la rama de la inteligencia artificial la misma que ha evolucionado en el análisis de datos para la ayuda en la toma de decisiones.

De ahí que se considere a la rama de la inteligencia artificial como medio para el reconocimiento de objetos, se ha propuesto implementar un algoritmo que reconozca objetos determinados.

La detección de objetos se puede utilizar para diversos fines ya sea en el ámbito social, político, tecnológico, industrial, económico, empresarial etc.[5] . Dentro del ámbito de Sistemas inteligentes, la detección de objetos en tiempo real.

Para los seres humanos es fácil reconocer o diferenciar las diversas formas de objetos que percibimos del mundo real, al contrario de un programa computarizado que su detección será más compleja puesto que se necesita de realizar funciones específicas para dicho reconocimiento en tiempo real.

En este trabajo se hará uso de todos los mecanismos, técnicas que permitan llevar a cabo el control de un brazo robótico para el reconocimiento de objetos. Una vez ingresados los objetos predeterminados a través de una cámara web en tiempo real, se realizará las respectivas pruebas de evaluación que permitan corroborar el reconocimiento de objetos.

El alcance de esta propuesta tecnológica es obtener un prototipo que realice la el reconocimiento de objetos en tiempo real.

1.2. Establecimiento de requerimientos

Este proyecto está diseñado y será ejecutado por fases, las cuáles se mencionan a continuación:

- Recolección de información, en esta fase, debemos hacer un estudio minucioso de la información obtenida y con ello contextualizar la realidad de la detección de objetos en tiempo real, así como también conocer los diversos materiales que se requerirán para la elaboración de nuestra propuesta tecnológica.
- Brazo robótico armado y funcional con los servomotores que permitan el desplazamiento de objetos.
- Establecer el script en Python en el cual se indica los parámetros para el reconocimiento de objetos.
- Pruebas o Testing, que permitan corroborar el correcto funcionamiento del script para la detección de objetos en tiempo real.

1.3. Justificación de requerimiento a satisfacer

El presente proyecto está dirigido al área de investigaciones Tecnológicas aplicadas a los procesos productivos para la producción sustentable establecido por la UTMACH.

La visión inteligente es aplicada en diversos campos y para diversos fines, como sabemos nuestro entorno posee una gran variedad de objetos que son fácilmente reconocidos por el ojo humano sin embargo aplicados al área de la inteligencia artificial este reconocimiento puede ser utilizado para diferenciar objetos animados de objetos inanimados, además de servir de base para ayuda pedagógica en el área educativa o como fuente de información para un estudio de marketing que les permita a las empresas conocer los objetos preferentes de sus cliente.

El prototipo a desarrollar hará uso de un brazo robótico el cual será el que seleccionará el objeto identificado en tiempo real mediante un script en Python y lo colocará en la clasificación correspondiente.

2. DESARROLLO DEL PROTOTIPO

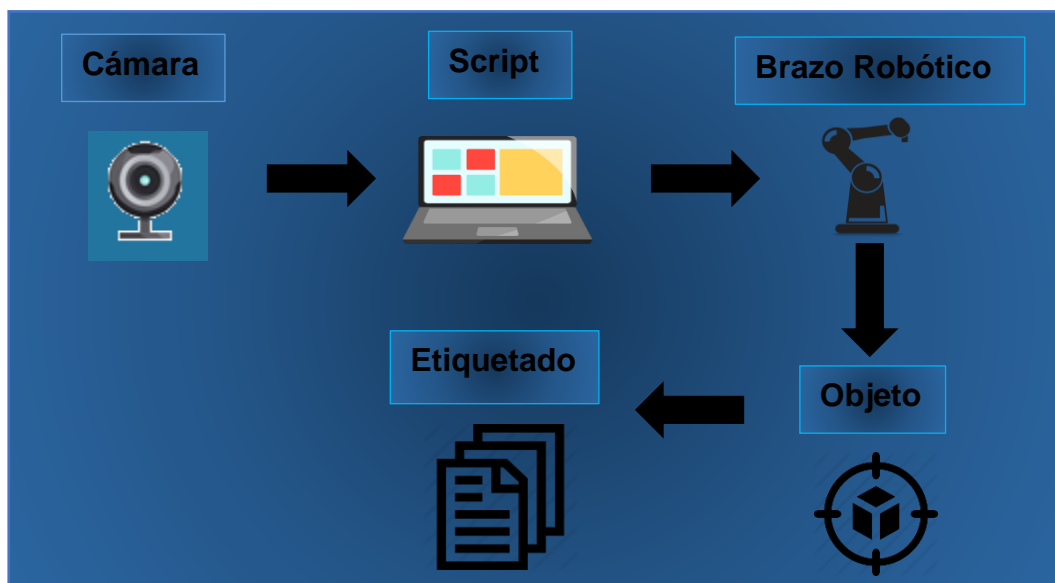
2.1. Definición del prototipo tecnológico

La aplicación de visión inteligente en el reconocimiento de objetos nos proporcionara como base de información que se puede aplicar a diferentes áreas para nuestro caso haremos uso de un brazo robótico[6] con servomotores los mismos que permitirán el movimiento de un objeto al ser reconocido.

Para el script del reconocimiento de objetos se hará uso de OpenCv[7] y Python [8] estas herramientas proporcionan un entorno que se adapta a las necesidades y requerimientos establecidos.

Las pruebas de comprobación para la detección de objetos se realizarán definiendo un número de objetos que serán identificados y clasificados de acuerdo a las clases previamente establecidas.

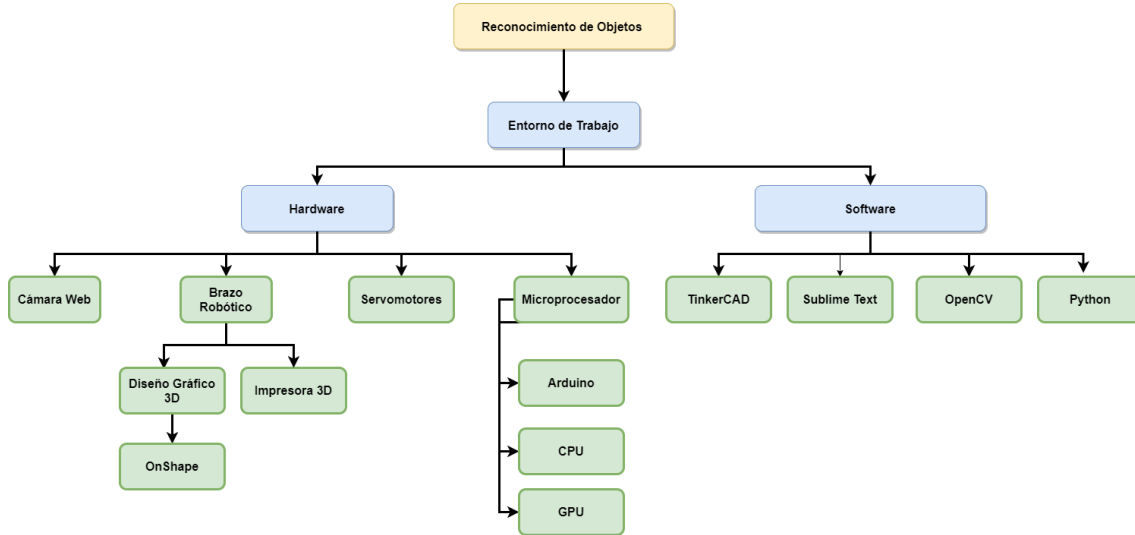
Figura 1: Arquitectura del Prototipo



Fuente: Elaboración propia

2.2. Fundamentación teórica del prototipo.

Figura 2: Mapa Mental de la Fundamentación Teórica del Prototipo



Fuente: Elaboración propia

2.2.1. Entorno de Trabajo

2.2.1.1. Software

2.2.1.1.1. Diseño Gráfico 3D

2.2.1.1.1.1. OnShape

Permite realizar diseños en 3D[9] como productos y maquinaria en muchas industrias, incluida la electrónica de consumo, maquinaria mecánica, dispositivos médicos a través de un navegador web o sistema operativo, además de sistema CAD de Onshape permite que varios usuarios accedan y trabajen en un solo diseño simultáneamente en la nube, utilizando cualquier computadora, tableta o teléfono.[10] Esta herramienta nos permitirá hacer el diseño de cada una de las piezas para el brazo robótico.

2.2.1.1.2. TinkerCAD

Es un programa de modelado 3D[11] en línea gratuito que se ejecuta en un navegador web es conocido por su simplicidad y facilidad de uso, se ha

convertido en una plataforma popular para crear modelos para impresión 3D, así como una introducción de nivel de entrada a la geometría sólida constructiva en las escuelas.[12]

2.2.1.1.3. OpenCv

La biblioteca OpenCv (Open Source Computer Vision) se puede utilizar tanto para uso académico como comercial. Esta biblioteca tiene interfaces en los lenguajes de programación C ++, C, Python y Java y es compatible con diferentes sistemas operativos.[13] OpenCv se integra perfectamente con el lenguaje de programación Python, en conjunto se logran obtener excelentes resultados, por lo que ha llegado a ser una de las librerías más usadas en lo que respecta a visión artificial, dada la efectividad y resultados obtenidos sobre procesamientos en tiempo real.[14]

2.2.1.1.4. Python

La programación es una habilidad esencial requerida en todos los campos del conocimiento, ya que puede aplicarse para dar solución a diversos problemas mediante el uso de computadoras y algoritmos. Python[15] ofrece un alto nivel de abstracción de la programación y se ha convertido en una herramienta útil para los científicos e informática científica en los últimos años debido a dos bibliotecas importantes: Numpy y Scipy.[16]

2.2.1.1.4.1. Numpy

Es una librería que pertenece a la biblioteca de librerías matemáticas la cual se dedica al cálculo sobre vectores y matrices en Python. Esta librería está escrita en lenguaje C la cual es eficiente al momento de trabajar con grandes cantidades de datos. La librería Numpy alcanzado una importancia muy alta que es indispensable para el ámbito de la Ciencia de Datos.[17]

2.2.1.2. Capacidad Computacional

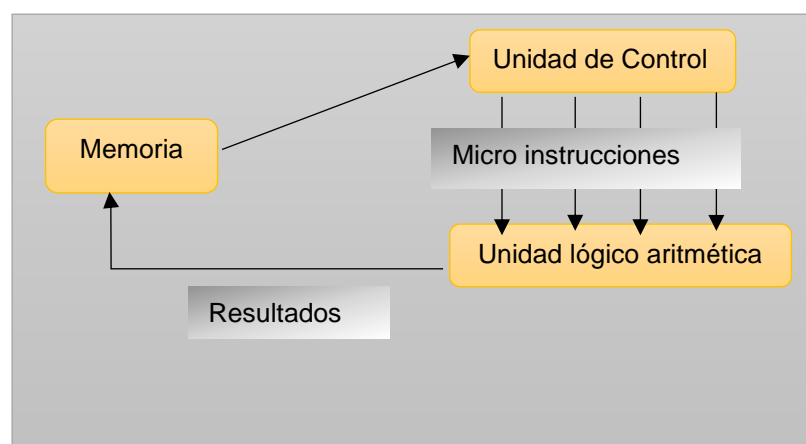
El aumento que está experimentando la sociedad en el ámbito tecnológico ha impulsado en la parte educativa a generar iniciativas que involucren actividades

que fomenten las habilidades asociadas a la programación y el pensamiento computacional como parte del mejoramiento y automatización a las diversas actividades que realiza la sociedad.[18] De ahí la necesidad de fomentar la investigación en cada área que ayude a mejorar la realización o automatización de procesos puesto que la rama computacional se puede incorporar o adaptar a las demás áreas.

2.2.1.2.1. CPU

Permite mejorar el rendimiento de las computadoras o dispositivos electrónicos. El CPU (Unidad Central de proceso) es uno de los componentes principales de una computadora la cual se encarga de procesar los datos y realizar cálculos. El CPU cuenta con una ALU (Arithmetic Logic Unit), una DRAM y una Cache.[19]

Figura 3: Estructura de un CPU



Fuente: Elaboración propia

2.2.1.2.2. GPU

GPU La GPU (Graphics Processing Unit) es uno de los componentes principales de una tarjeta gráfica la cual contiene una serie de millones de transistores que realizarán los cálculos y procesos[20]. A diferencia de la CPU la GPU cuenta con varias ALUs, varios Controles, varios Caches y una DRAM[21]. Su principal uso fue originalmente para procesar imágenes en alta calidad y videojuegos, hoy en día es una herramienta indispensable dentro de la visión inteligente.

2.2.1.2.3. Cámara Web

Se utiliza una cámara web[22] o sensor que permite capturar una imagen y transmitirla al computador donde se procesara e identifica además de clasificarla en la categoría correspondiente, con esto se pretende facilitar el funcionamiento del algoritmo para la detección del objeto[23]. La cámara con puerto USB 2.0 posee las siguientes características: resolución de 2 Mpx,[24]

2.2.1.2.4. Brazo Robótico

La tecnológica ha evolucionado exponencialmente y podemos evidenciar en la parte robótica [25] con la creación de dispositivos que automaticen procesos, tareas o actividades que realiza el ser humano.[26] Los sistemas robóticos cada vez son más complejos y las teorías de la ley de control tradicional pierden robustez, debido al aumento de requerimiento y con ello también aumenta la dificultad con la que se puede controlar el robot para interactuar con el entorno que lo rodea.[27]

2.2.1.2.5. Impresora 3D

Es una impresora 3D que ofrece alta calidad de impresión y al mismo tiempo fácil de usar[28], posee dos extrusores que pueden imprimir el mismo material y crear impresiones en dos colores o incluso imprimir utilizando dos materiales diferentes a la vez. También utiliza hardware propietario propio para sus impresoras, y con ello asegurarse de que su sistema puede funcionar como se pretende.[29]

2.2.1.2.6. Servomotores

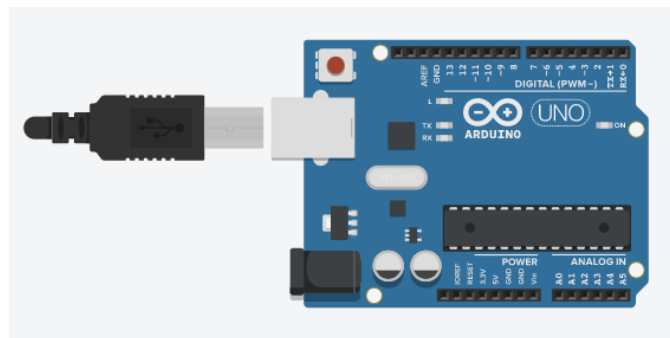
Los servomotores de modelismo operan a voltajes bajos en corriente directa entre 4 y 6 voltios, son controlados mediante el envío de impulsos eléctricos de Arduino. Estos pulsos le dicen al servomotor a qué posición se debe mover.[30] Entre los servomotores podemos encontrar:

Servomotores de rotación continua: se caracterizan por ser capaces de girar 360 grados, es decir, una rotación completa se puede controlar su posición y velocidad de giro en un momento dado.

2.2.1.2.7. Arduino

Es una creación electrónica de código abierto cuyo uso es fácil para los creadores y desarrolladores.[31] Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso.[32] La tarjeta Arduino es usada para procesar los datos y con ello poder controlar los movimientos del prototipo funcional robótico desarrollado para realizar la selección de objetos.[33]

Figura 4: Arduino



Fuente: [9]

2.2.2. Técnicas de Tratamiento de Imágenes

2.2.2.1. Detección de Objetos

El ser humano puede observar una imagen e inmediatamente reconocerla e identificar las características del objeto que es, al contrario de una computadora que requiere de algoritmos visión inteligente[34] para identificar características previamente establecidas.

2.2.3. Algoritmos de detección de objetos

2.2.3.1. Sublime text

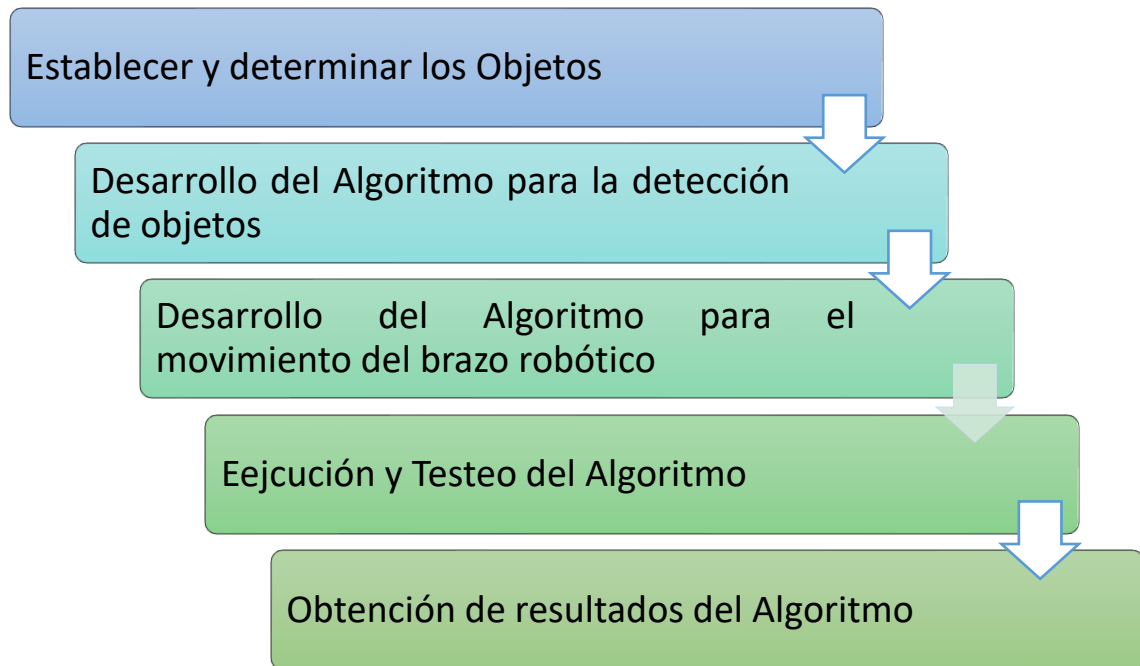
Sublime Text es un editor de texto y editor de código fuente[35]. Está escrito en C++ y Python para los plugins. Desarrollado originalmente como una extensión de Vim, es una herramienta de fácil uso y que se adapta a nuestras necesidades básicas para la programación.[36]

2.3. Metodología

Posterior a la recopilación de información se analizó y con ello identificó técnicas para la detección de objetos en base a una clasificación de objetos. El objetivo principal es realizar el reconocimiento de objetos haciendo uso de técnicas de aprendizaje profundo mediante visión inteligente.

Primero se realizó la clasificación de los objetos que utilizaremos como base para el reconocimiento de objetos al cual se le aplico aprendizaje profundo utilizando Python. Luego se realizó pruebas de ejecución y testeo para obtener métricas q permitan a través de matrices de confusión obtener resultados porcentuales en cuanto al reconocimiento de objetos.

Figura 5: Metodología



Fuente: Elaboración propia

2.4. Objetivos del Prototipo

2.4.1. Objetivo General

Implementar un sistema para el reconocimiento de objetos aplicando algoritmos con visión inteligente

2.4.2. Objetivo Especifico

- Investigar fuentes científicas relacionados con la detección de objetos mediante visión inteligente.
- Construir un modelo eficiente que permita el reconocimiento de objetos.
- Realizar pruebas de detección en tiempo real para el reconocimiento de objetos.
- Obtener métricas de evaluación de las pruebas a través de una matriz de confusión.

2.5. Diseño del prototipo

El diseño del prototipo se divide en tres fases: primero identificamos todos los materiales (hardware y software) que se utilizaran para la realización del sistema, la segunda fase consiste en la elaboración del algoritmo para el reconocimiento de objetos mediante la búsqueda de información y del lenguaje a utilizar, por último, se realizan pruebas que nos permitan determinar si el sistema establecido cumple las necesidades requeridas.

A continuación, se detalla los recursos a utilizar:

2.5.1. Software

Tabla 1: Software Requerido

| Software (paquetes, librerías, etc) | Versión |
|-------------------------------------|---|
| Python | 3.7.3 |
| OpenCv | 3.4 o superior |
| Sublime Text | 2 o superior |
| onShape | 1.106.7080 o superior (disponible online) |
| TinkerCAD | Disponible online |

Fuente: Elaboración propia

2.5.2. Requisitos Hardware

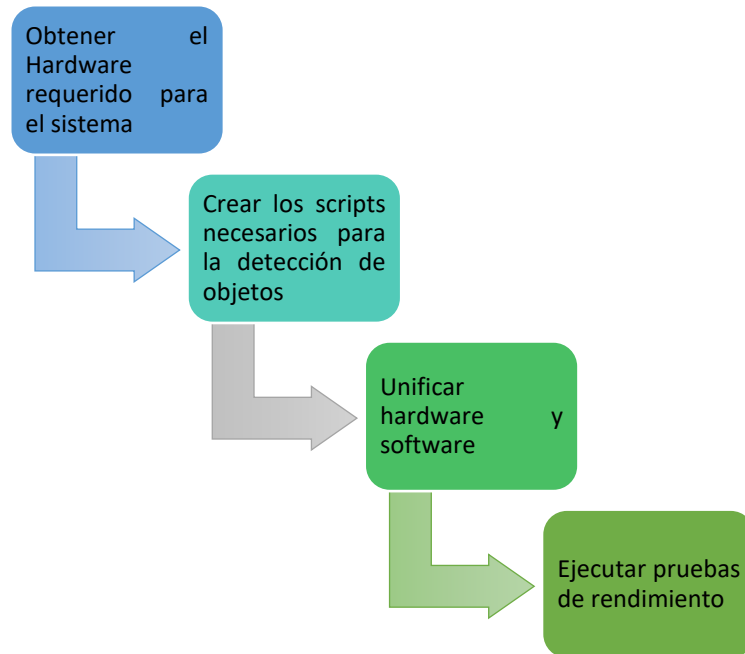
Tabla 2: Hardware Requerido

| Hardware Requerido |
|---|
| Marca: HP |
| Procesador: Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz |
| Memoria: 16,0 GB |
| Sistema Operativo: Windows 10 |

Fuente: Elaboración propia

2.5.3. Obtención del modelo para el reconocimiento de Objetos

Figura 6: Flujo de Desarrollo del Sistema



Fuente: Elaboración propia

2.6. Ejecución y/o ensamblaje del prototipo

2.6.1. Obtener el hardware Requerido

Para el desarrollo del sistema haremos uso de una cámara web, un brazo robótico, servomotores, una tarjeta Arduino, cables para las conexiones.

Figura 7: Hardware Requerido



Fuente: Elaboración propia

2.6.2. Crear los Scripts necesarios para la detección de Objetos

2.6.2.1. Scripts para la detección de Objetos

1. Determinar los objetos que servirán como base para el reconocimiento en tiempo real.
2. Crear una carpeta que contendrá los scripts.
3. Crear un archivo *FigurasGeometrica.py*, importar las librerías necesarias.
4. Establecer las variables e inicializarlas.
5. Cargar el modelo en memoria.
6. Procesamiento de la imagen de entrada.
7. Identificar el objeto ingresado por imagen.
8. Verificar si el reconocimiento es el correcto.

Nota: El código fuente para esta sección se encuentra en el Anexo I.

2.6.2.2. Scripts para la selección del objeto requerido para la tarjeta Arduino

1. Determinar el número de servomotores a utilizar que permitan el movimiento del brazo robótico.
2. Crear un archivo *AplicaciónCV.ino*.
3. Inicializar variables requeridas.
4. Procesamiento si detecta el objeto el sensor se establece en 1 caso contrario en 0.
5. Crear los métodos necesarios que permitan el movimiento para cada caso establecido.
6. Ejecutar el script.
7. Verificar si el resultado es el esperado.

Nota: El código fuente para esta sección se encuentra en el Anexo II.

3. EVALUACIÓN DEL PROTOTIPO

3.1. Plan de evaluación

De los resultados obtenidos se efectúa un análisis comparativo, que permitan verificar la predicción del reconocimiento de objetos. Se realizará una prueba con un número determinado de objetos y con ello comprobar si dichos resultados satisfacen los requerimientos previamente establecidos.

3.1.1. Métricas de Clasificación

Una matriz de confusión o matriz de error, es una tabla de contingencia que sirve como herramienta estadística para el análisis de observaciones emparejadas[37]. La matriz de confusión se utiliza en muchos casos como un estándar para informar sobre la exactitud temática de cualquier producto de datos derivados de la teledetección.

La matriz de confusión se basará en los siguientes parámetros:

- VP: la cantidad de positivos que el modelo clasificó correctamente como positivos. •
- VN: la cantidad de negativos que el modelo clasificó correctamente como negativos. •
- FN: la cantidad de positivos que el modelo clasificó incorrectamente como negativos. •
- FP: la cantidad de negativos que el modelo clasificó incorrectamente como positivos.

Tabla 3: Matriz de Confusión

| | | Predicción | |
|----------|----------|-------------------------|-------------------------|
| | | Positivo | Negativo |
| Realidad | Positivo | Verdadero Positivo (VP) | Falso Negativo (FN) |
| | Negativo | Falso Positivo (FP) | Verdadero Negativo (VN) |

Fuente: Elaboración propia

A partir de esta matriz se puede calcular la precisión, sensibilidad, puntuación de referencia, exactitud, etc.

Exactitud

La exactitud (accuracy) es el porcentaje de predicciones que fueron correctas.[37]

$$Accuracy = \frac{tp+tn}{tp+fp+tn+fn}$$

Precisión

La precisión es el porcentaje de exactitud en los resultados de la predicción. Su fórmula es el total de los verdaderos positivos dividido entre la suma de los verdaderos positivos y falsos positivos.[37]

$$Precision = \frac{tp}{tp+fp}$$

Sensibilidad (Recall) Es la efectividad que tiene el modelo para identificar etiquetas positivas. Su fórmula se calcula dividiendo el número de verdaderos positivos entre la suma de los verdaderos positivos y los falsos negativos.[38]

$$Recall = \frac{tp}{tp+fn}$$

Puntaje de Referencia (F-Score)

Es una mezcla de precisión y sensibilidad, nos da una referencia de las relaciones entre resultados positivos del modelo. Se calcula multiplicando 2 por la precisión por la sensibilidad y dividiendo este resultado para la suma de precisión y sensibilidad.[38]

$$F - Score = 2 * \frac{Precision*Recall}{Precision+Recall}$$

3.2. Resultado de evaluación

3.2.1. Resultado de la Prueba de Entrenamiento y Validación

Se estableció realizar una prueba con número determinado de objetos (figuras geométricas) para determinar el grado de efectividad que tiene el algoritmo realizado para la detección de objetos en tiempo real. Así como determinar los parámetros adecuados que permitan el reconocimiento e identificación de dichos objetos.

Prueba 1. Considerando un número de 50 objetos

Una vez ejecutado el algoritmo para el reconocimiento de objetos podemos observar en las figuras los datos obtenidos:

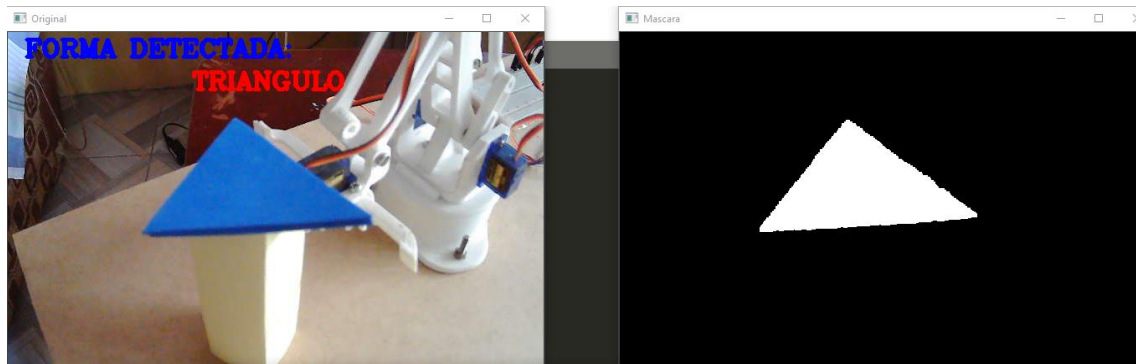
Tabla 4: Información de la Prueba 1

| Información de entrada | | |
|------------------------------|---|-----------|
| Entrada | Imágenes de objetos (figuras geométricas) | |
| Detecciones | Segura | No Segura |
| Detección del algoritmo | 43 | 7 |
| Total, de objetos detectados | 43 | |
| Total, de objetos | 50 | |

Fuente: Elaboración propia

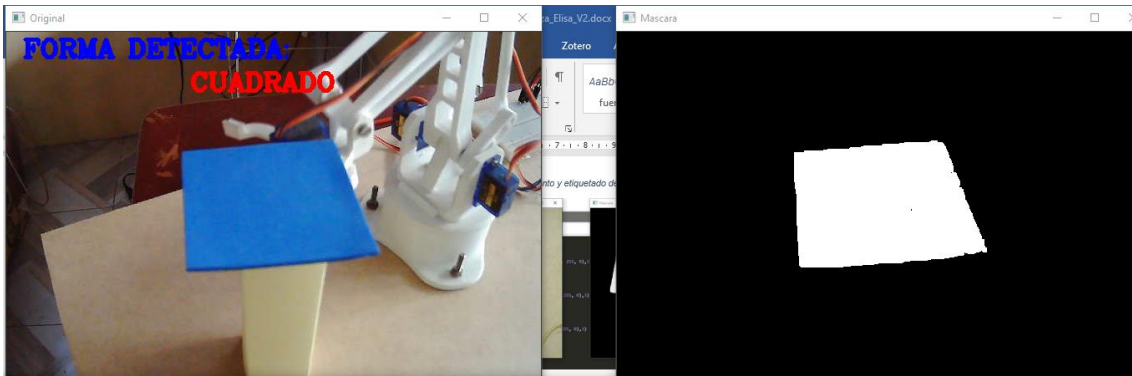
La detección de objetos es aceptable se logró identificar 43 de 50 existentes demostrando que existe un número de objetos (figuras geométricas) que no se reconocen.

Figura 8: Reconocimiento y etiquetado del objeto (Triángulo)



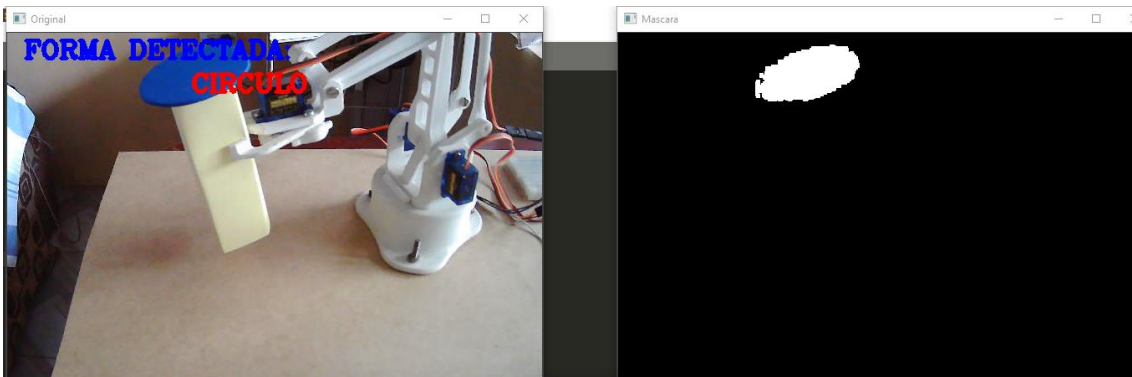
Fuente: Elaboración propia

Figura 9: Reconocimiento y etiquetado del objeto (cuadrado)



Fuente: Elaboración propia

Figura 10: Reconocimiento y etiquetado del objeto (circulo)



Fuente: Elaboración propia

Tabla 5: Matriz de Confusión Prueba 1

| | | Predicción | |
|-------------------|--------------------------|-----------------------|--------------------------|
| | | Reconocimiento Seguro | Reconocimiento no seguro |
| Realidad | Reconocimiento seguro | 43 | 7 |
| | Reconocimiento no seguro | 4 | 3 |
| Resultados | | | |
| Precisión | | 0.9047 | |

| | |
|---------------------|--------|
| Sensibilidad | 0.8837 |
| Exactitud | 0.8200 |
| Valor de Referencia | 0.4470 |

Fuente: Elaboración propia

Se obtuvo una precisión de 90% y una sensibilidad de 88%. Esto significa que el algoritmo detecta considerablemente bien los objetos en tiempo real siempre y cuando la cámara este ubicada en un ángulo de distancia que permita visualizar los objetos correctamente.

3.3. Conclusiones

Una vez realizado las pruebas para el reconocimiento de objetos haciendo uso de los algoritmos se concluye que:

- Dentro de las fuentes de información no hay muchas implementaciones relacionadas al reconocimiento de objetos (figuras geométricas: círculo, cuadrado, triángulo) en tiempo real haciendo uso de un brazo robótico, lo cual resulta hasta cierto punto un impedimento para el respaldo bibliográfico al ser temas que se traten de forma independiente.
- Desarrollar un algoritmo para el reconocimiento de objetos es bastante amplio puesto que se debe considerar las formas además de tratar de enmascarar el entorno y que únicamente reconozca el objeto predeterminado (figura geométrica) mediante una cámara.
- Los métodos de visión inteligente nos ofrecen una forma óptima de reconocer objetos a través de una imagen ingresada en tiempo real.
- Se pudo obtener un buen rendimiento del algoritmo de reconocimiento de objetos a partir de una imagen en tiempo real.
- Las pruebas obtenidas en la ejecución del prototipo evidenciaron porcentajes de precisión bastante buenos en cuanto a los resultados de las métricas obtenidas.

3.4.Recomendaciones

En base a los resultados del algoritmo de reconocimiento de objetos se recomienda:

- Profundizar en técnicas y métodos para el reconocimiento de objetos y con ello lograr un mejor desempeño del sistema en base a valores más realistas.
- Investigar métodos de calibración de cámara y otros que permitan determinar la profundidad de una imagen para obtener mejores cálculos para el reconocimiento de objetos.
- Hacer varias pruebas con imágenes para el reconocimiento de objeto correcto para determinar una mejor aproximación a la cantidad de objetos reconocidos.
- Interpretar correctamente los resultados de las métricas de evaluación del modelo para con ello mejorar la calidad del reconocimiento de objetos y la fiabilidad de los resultados.

4. BIBLIOGRAFÍA

[1] R. D. Moreno Padilla, «La llegada de la inteligencia artificial a la educación», *RITI*, vol. 7, n.º 14, pp. 260-270, 2019, doi: <https://doi.org/10.36825/RITI.07.14.022>.

[2] A. Alarcón Peña, J. C. Villalba Cuéllar, y J. F. Franco Mongua, «La inteligencia artificial y su impacto en la enseñanza y el ejercicio del derecho», *Prolegómenos*, vol. 22, n.º 44, p. 4, 2019, doi: <https://doi.org/10.18359/prole.4353>.

[3] J. L. Aznar Lahoz, «La robótica en la guerra del futuro», *bie3 IEEE*, n.º 17, pp. 781-812, 2020.

[4] C. Agulló, «Reconocimiento geométrico de objetos 3D y detección de deformaciones en manipulación robótica», Universidad de Alicante, España, 2017.

[5] X. Aguas, M. Herrera, N. Sotomayor, y O. Camacho, «Prototipo de robot paralelo accionado por cuatro cables: resultados experimentales», *UTE*, vol. 10, n.º 1, p. 14, 2019, doi: DOI: 10.29019/enfoqueute.v10n1.440.

[6] B. I. Asqui Ramírez, E. F. García Cabezas, J. M. Orozco Ramos, y J. C. Cayán Martínez, «Control y monitoreo de brazo robótico integrando Labview, Solidworks y la internet de las cosas», *Koinonia*, vol. 5, n.º 2, pp. 30-47, 2020, doi: <http://dx.doi.org/10.35381/r.k.v5i2.874>.

[7] S. Sivkov *et al.*, «The algorithm development for operation of a computer vision system via the OpenCV library», *Procedia Computer Science*, vol. 169, pp. 662-667, 2020, doi: <https://doi.org/10.1016/j.procs.2020.02.193>.

[8] J. R. Ruiz Sarmiento, J. Monroy, F. A. Moreno, y J. Gonzalez Jimenez, «Tutorial para el reconocimiento de objetos basado en características empleando herramientas python», en *Badajoz*, 5-7, España, 2018, pp. 998-1005, doi: <https://doi.org/10.17979/spudc.9788497497565.0998>.

- [9] P. X. Ramírez Torres, A. I. Paredes Amaguaya, X. P. Bejarano Gavilanes, y M. L. Pomaquero Yuquilema, «Análisis de tendencias de diseño gráfico aplicadas en diseño web», *Dominio de las Ciencias*, vol. 6, n.º 3, pp. 923-932, 2020, doi: <http://dx.doi.org/10.23857/dc.v6i3.1257>.
- [10] «Onshape es la plataforma de desarrollo de productos todo en uno completa.», *Tecnológica*, 2014. <https://www.onshape.com/en/platform> (accedido abr. 05, 2021).
- [11] G. C. Galiano Murillo, J. A. Madrigal Lobo, W. Gutiérrez Sandí, y T. Lemos Pires, «Aplicaciones de las Impresoras 3D en el desarrollo de prótesis para cirugía de reemplazo protésico (artroplastias)», *Tecnología Marcha*, vol. 31, n.º 3, pp. 131-142, 2018, doi: DOI: 10.18845/tm.v31i3.3908.
- [12] «Autodesk tinkercad», *tinkercad*, 2021. <https://www.tinkercad.com/> (accedido abr. 05, 2021).
- [13] J. de Fátima da Silva, C. Bertolini, F. J. Parreira, y S. R. Silveira, «Protótipo de um Aplicativo Mobile para Reconhecimento de Vestimenta para Pessoas Cegas», *Educação Especial*, vol. 32, pp. 1-18, 2019, doi: <https://doi.org/10.5902/1984686X32551>.
- [14] M. A. Palma Jaramillo, R. V. Vaca Moscoso, Y. M. Torres Berru, D. E. Granda Gutiérrez, y L. F. León Pinzón, «Diseño de un prototipo para cobro de peajes con visión artificial», *Polo del Conocimiento*, vol. 5, n.º 7, pp. 60-76, 2020, doi: 10.23857/pc.v5i7.1495.
- [15] C. A. Altamirano Loor, L. O. Albarracín Zambrano, y D. V. Ponce Ruiz, «Sistema inteligente para el reconocimiento de figuras geométricas basado en Python con Raspberry Pi», *KOINONIA*, vol. 5, n.º 2, pp. 18-29, 2020, doi: <http://dx.doi.org/10.35381/r.k.v5i2.872>.
- [16] O. Solarte Pabón, «Fomentar la motivación y mejorar el rendimiento de los estudiantes en un curso de introducción a la programación: un enfoque de enseñanza integrado», *EIA*, vol. 16, n.º 31, pp. 65-78, 2019, doi: <https://doi.org/doi.org/10.24050/reia.v16i31.1230>.

- [17] J. Prieto Villamar, L. Apupalo Del Rosario, M. Quiroz Martínez, y G. Valverde Landivar, «Detección de personal no autorizado en el departamento de ti utilizando redes neuronales convolucionales en tiempo real con raspberry pi 3 b+», *journal of science and research*, vol. 5, n.º 3, pp. 49-60, 2020, doi: <https://doi.org/10.5281/zenodo.3926937>.
- [18] Y. A. Caballero González y G. V. Muñoz Repiso, «Robots en la educación de la primera infancia», *RIED*, vol. 24, n.º 1, pp. 77-94, 2021, doi: <https://doi.org/10.5944/ried.24.1.27508>.
- [19] C. Marlis Fulgueira, E. Insúa Suárez, y H. Díaz Pando, «Implementación del Algoritmo Trace Alignment Empleando Técnicas de Programación Paralela», *Lámpsakos*, n.º 15, pp. 11-21, 2016, doi: <http://doi.org/>.
- [20] A. Lopez Fernandez, D. Rodriguez Baena, F. Gomez Vela, F. Divina, y M. Garcia Torres, «Un algoritmo de biclustering de múltiples GPU para conjuntos de datos binarios», *Journal of Parallel and Distributed Computing*, vol. 147, pp. 209-219, 2020, doi: <https://doi.org/10.1016/j.jpdc.2020.09.009>.
- [21] F. J. Enríquez Aguilera, J. M. Silva Aceves, S. V. Torres Argüelles, E. A. Martínez Gómez, y G. Bravo Martínez, «Utilización de GPU-CUDA en el Procesamiento Digital de Imágenes», *Electrónicas*, vol. 15, n.º 66, pp. 65-79, 2018, doi: <http://dx.doi.org/10.20983/culcyt.2018.3.9>.
- [22] D. Delgado León, «Diseño de un sistema de adquisición de imágenes basado en cámaras web USB y hardware reconfigurable», *RIELAC*, vol. 38, n.º 2, pp. 1-11, 2017.
- [23] S. Amaya Zapata, D. Pulgarín Velásquez, y I. D. Torres Pardo, «Desarrollo e Implementación de un Sistema de Visión Artificial Basado en Lenguajes de Uso Libre para un Sistema Seleccionador de Productos de un Centro Integrado de Manufactura (CIM)», *Lámpsakos*, n.º 15, pp. 43-50, 2016, doi: <http://dx.doi.org/10.21501/21454086.1702>.
- [24] F. Hubert Sánchez, T. Vázquez Romaguera, y C. R. Vázquez Seisdedos, «Sistema de estimulación y registro del movimiento ocular con el empleo de la

videoculografía infrarroja», *MEDISAN*, vol. 24, n.º 3, p. 12, 2020, doi: <http://orcid.org/0000-0001-7013-1434>.

[25] J. J. Padilla Mora, «Diseño e implementación de un controlador no-lineal para un brazo robótico con articulaciones flexibles y enlaces rígidos», *Tecnología en Marcha*, vol. 33, n.º 2, pp. 160-168, 2020, doi: <https://doi.org/10.18845/tm.v33i6.5312>.

[26] K. Morales, C. Hoyos, y J. García Caicedo, «Diseño y optimización de la estructura mecánica de un brazo robótico antropomórfico desarrollado con fines educativos», *UIS*, vol. 18, n.º 4, pp. 193-208, 2019, doi: 0000-0001-5466-9429.

[27] J. Cesar Correa, «Aproximaciones al futuro de la robótica en Colombia», *Lámpsakos*, n.º 17, pp. 9-12, 2017, doi: 10.21501/21454086.2398.

[28] A. Ortiz Gil, «Las impresoras 3D como herramientas científicas», *Encuentros multidisciplinares*, vol. 21, n.º 61, p. 8, 2019.

[29] impresorad3d, «cube 3», *impresorad3d*. <http://impresorad3d.com/cube-3/>.

[30] L. C. Burbano Castillo, R. P. Trullo Daza, y G. Daraviña Peña, «Desarrollo de una interfaz gráfica para el control de posición de un manipulador paralelo plano 3RRR», *Scientia Et Technica*, vol. 23, n.º 2, pp. 168-174, 2018.

[31] E. V. Altamirano Santillan, G. E. Vallejo Vallejo, y J. C. Cruz Hurtado, «Manipulación de un servomotor con un módulo acelerómetro de 3 ejes MMA 7361 empleando “Arduino” y “Simulink”», *Dominio de las Ciencias*, vol. 3, n.º 2, pp. 1006-1030, 2017.

[32] M. Zambrano de la Torre, M. Guzmán Fernández, y C. Sifuentes Gallardo, «Una Guía Práctica para Desarrollar Equipo de Laboratorio con Arduino», *Conciencia Tecnológica*, vol. 1, n.º 59, p. 14, 2020.

[33] E. Ceballos, M. Díaz Rodríguez, J. L. Paredes, y P. Vargas, «Desarrollo de un Robot de Rehabilitación pasiva para la articulación de la muñeca mediante la implementación de un microcontrolador Arduino UNO», *UIS*, vol. 16, n.º 1, p. 20, 2017.

- [34] «Segmentación y detección de objetos en imágenes y vídeo mediante inteligencia computacional», Tesis Doctoral, Universidad de Málaga, España, 2018.
- [35] J. Juliana Plaza Santillan, M. Bueno Salinas, y E. Delgado Saeteros, «Los Sitios Web y su Incidencia en el Desarrollo de la Gestión Documental», *Generando Productividad Institucional*, vol. 6, n.º 3, p. 6, 2018, doi: <https://doi.org/10.34070/rif.v7i1>.
- [36] «sublimetext», *Un editor de texto sofisticado para código, marcado y prosa*, 2020. <https://www.sublimetext.com/>.
- [37] F. J. Ariza López, J. Rodríguez Avi, y A. Fernández, «Control estricto de matrices de confusión por medio de distribuciones multinomiales», *Ciencia y tecnología de la Información Geográfica*, vol. 1, n.º 21, pp. 215-226, 2018, doi: <http://dx.doi.org/10.21138/GF.591>.
- [38] D. M. Garcés Eslava, «Método de procesamiento de lenguaje natural y técnicas de minería de datos aplicadas a la clasificación de incidentes informáticos», *INTERFACES*, vol. 1, n.º 12, pp. 11-29, 2019, doi: [doi: 10.26439/interfases2019.n012.4635](https://doi.org/10.26439/interfases2019.n012.4635).

5. ANEXOS

5.1. Anexo I. Código Fuente archivo “figurasGeometricas.py”

```
1 import cv2
2 import numpy as np
3 import serial, time
4 arduino = serial.Serial("COM4", 9600)
5 time.sleep(2)
6
7
8 def nothing(x):
9     # any operation
10    pass
11
12 cap = cv2.VideoCapture(0)
13
14 cv2.namedWindow("Trackbars")
15 cv2.createTrackbar("L-H", "Trackbars", 72, 180, nothing)
16 cv2.createTrackbar("L-S", "Trackbars", 193, 255, nothing)
17 cv2.createTrackbar("L-V", "Trackbars", 255, 255, nothing)
18 cv2.createTrackbar("U-H", "Trackbars", 180, 180, nothing)
19 cv2.createTrackbar("U-S", "Trackbars", 255, 255, nothing)
20 cv2.createTrackbar("U-V", "Trackbars", 255, 255, nothing)
21
22 font = cv2.FONT_HERSHEY_COMPLEX
23 valor = b'0'
24 listo = b'0'
25 figura = "NINGUNA"
26
27 while True:
28     _, frame = cap.read()
29     hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
30
```

```
31     l_h = cv2.getTrackbarPos("L-H", "Trackbars")
32     l_s = cv2.getTrackbarPos("L-S", "Trackbars")
33     l_v = cv2.getTrackbarPos("L-V", "Trackbars")
34     u_h = cv2.getTrackbarPos("U-H", "Trackbars")
35     u_s = cv2.getTrackbarPos("U-S", "Trackbars")
36     u_v = cv2.getTrackbarPos("U-V", "Trackbars")
37
38     lower_red = np.array([l_h, l_s, l_v])
39     upper_red = np.array([u_h, u_s, u_v])
40
41     mask = cv2.inRange(hsv, lower_red, upper_red)
42     kernel = np.ones((5, 5), np.uint8)
43     mask = cv2.erode(mask, kernel)
44     mask = cv2.dilate(mask, kernel)
45
46     if (arduino.inWaiting()):
47         listo = arduino.read()
48         print("llego ", listo)
49         # Contours detection
50         contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
51
52         for cnt in contours:
53             area = cv2.contourArea(cnt)
54             approx = cv2.approxPolyDP(cnt, 0.02*cv2.arcLength(cnt, True), True)
55             x = approx.ravel()[0]
56             y = approx.ravel()[1]
57
58             if area > 400:
59                 cv2.drawContours(frame, [approx], 0, (0, 0, 255), 4)
60
```

```

61         if (len(approx) == 3):
62
63             cv2.putText(frame, "TRIANGULO", (x, y), font, 1, (0, 255, 0),3)
64             figura = "TRIANGULO"
65             valor=b'1'
66
67         elif (len(approx) == 4):
68
69             cv2.putText(frame, "RECTANGULO", (x, y), font, 1, (0, 255, 0),3)
70             figura = "RECTANGULO"
71             valor=b'2'
72
73         elif (7 < len(approx) < 20):
74
75             cv2.putText(frame, "CIRCULO", (x, y), font, 1, (0, 255, 0),3)
76             figura = "CIRCULO"
77             valor=b'3'
78
79     if listo == b'1':
80         print("===entro ",valor)
81         arduino.write(valor)
82         valor = b'0'
83         listo = b'0'
84     cv2.putText(frame, "FORMA DETECTADA: ", (20, 30), font, 1, (255, 0, 0),3)
85     cv2.putText(frame, figura, (220, 70), font, 1, (0, 0, 255),3)
86     cv2.imshow("Original", frame)
87     cv2.imshow("Mascara", mask)
88
89

```

```

90
91
92
93     key = cv2.waitKey(1)
94     if key == 27:
95         break
96
97     cap.release()
98     cv2.destroyAllWindows()
99     arduino.close()
100

```

5.2. Anexo I. Código fuente archivo “AplicaciónCV.ino”

```
#include <Servo.h>
#include "ServoEaser.h"

Servo servo_5;
Servo servo_6;
Servo servo_9;
Servo servo_10;
Servo servo_11;
Servo servo_12;

ServoEaser servo_5E;
ServoEaser servo_6E;
ServoEaser servo_9E;
ServoEaser servo_10E;
ServoEaser servo_11E;
int listo=1;
int servoFrameMillis = 10; // minimum time between servo updates

void setup() {
  Serial.begin(9600);
  attachInterrupt(digitalPinToInterrupt(2), sensor, FALLING); // el sensor que usa la cinta pasa de 1 a 0 cuando detecta un objeto
                                                                // se utiliza la interrupcion del pin 2

  servo_5.attach(5);
  servo_6.attach(6);
  servo_9.attach(9);
  servo_10.attach(10);
  servo_11.attach(11);
  servo_12.attach(12);

  servo_5E.begin( servo_5, servoFrameMillis );
  servo_6E.begin( servo_6, servoFrameMillis );
  servo_9E.begin( servo_9, servoFrameMillis );
  servo_10E.begin( servo_10, servoFrameMillis );
  servo_11E.begin( servo_11, servoFrameMillis );

  mover(90,90,90,90,90,2000);
  delay(5000);
  servo_12.write(105);
}

void loop() {
  if (listo ==1 ){
    delay(1000);
    listo =1;
    Serial.write('2'); //pido valor enviando un digito cualquiera
    while (Serial.available()==0){}
    char option = Serial.read();
    switch (option){
      case '1':
        tomarIri();
        break;
      case '2':
        tomarCua();
        break;
      case '3':
        servo_11.write(10);
```

```

        delay(2000);
        servo_11.write(60);
        delay(2000);
        servo_11.write(150);
        delay(2000);
        servo_11.write(5);
        delay(2000);
        break;
    default:
        servo_12.write(105);
        break;
    }
}

}

void sensor(){
    servo_12.write(90);
    listo = 1;
}

void tomar(){
    mover(111,54,111,30,90,1000);
    mover(120,45,105,30,90,1000);
    mover(120,45,105,30,150,500);
    mover(90,81,90,90,150,1000);
    servo_12.write(105);
}

void tomarTri(){
    tomar();
    mover(48,57,87,95,150,1000);
    mover(48,57,87,95,90,500);
    mover(90,81,90,90,90,1000);
}

void tomarCua(){
    tomar();
    mover(0,60,90,95,150,1000);
    mover(72,72,108,95,90,500);
    mover(90,81,90,90,90,1000);
}

```

```
void tomarCir(){
    tomar();
    mover(60,66,99,95,150,1000);
    mover(60,66,99,95,90,500);
    mover(90,81,90,90,90,1000);
}

void mover(int base,int hombro,int brazo,int mano,int garra,int tiempo){
    servo_5E.easeTo( base, tiempo );
    servo_6E.easeTo( hombro, tiempo );
    servo_9E.easeTo( brazo, tiempo );
    servo_10E.easeTo( mano, tiempo );
    servo_11E.easeTo( garra, tiempo );

    while( !servo_5E.hasArrived() or !servo_6E.hasArrived() or !servo_9E.hasArrived() or !servo_10E.hasArrived() or !servo_11E.hasArrived()){
        servo_5E.update();
        servo_6E.update();
        servo_9E.update();
        servo_10E.update();
        servo_11E.update();
    }
}
```