

UNIVERSIDAD SANTO TOMÁS

Sistema de identificación y recolección de productos agrícolas para el brazo robótico ReactorX200 basado en segmentación semántica.

Realizado por

Holman Danilo Nieto González

Proyecto de Grado presentado en cumplimiento del requisito parcial para optar por el grado de Ingeniería Electrónica



Grupo de Investigación GED (Grupo de Estudio y Desarrollo en Robótica)
Facultad de Ingeniería Electrónica
División de Ingenierías

Diciembre de 2024

Sistema de identificación y recolección de productos agrícolas para el brazo robótico ReactorX200 basado en segmentación semántica.

Realizado por

Holman Danilo Nieto González

Proyecto de Grado presentado en cumplimiento del requisito parcial para optar por el
grado de Ingeniería Electrónica

Dirigido por

Armando Mateus Rojas M.Sc

Codirigido por

Sindy Paola Amaya M.Sc

Grupo de Investigación GED (Grupo de Estudio y Desarrollo en Robótica)

Facultad de Ingeniería Electrónica

División de Ingenierías

Diciembre de 2024

Agradecimientos

Con la bendición de Dios y el apoyo incondicional de quienes me rodean, hoy culmino un capítulo importante en mi vida académica. A cada uno de ustedes, les debo un profundo agradecimiento por su respaldo y constancia durante este viaje de aprendizaje y crecimiento.

Primeramente Dios, quien me ha dado la fortaleza, la sabiduría y la perseverancia para realizar este trabajo, le doy mi más sincero agradecimiento. Su gracia ha sido mi guía constante a lo largo de cada desafío y victoria.

A mamá, cuyo amor incondicional, dedicación y paciencia han sido mi mayor inspiración, le agradezco infinitamente por su apoyo constante. Sus palabras de aliento y su ejemplo de fortaleza han sido la luz que ilumina mi camino y el lugar seguro al que puedo recurrir cada vez que me siento perdido.

A papá, cuyo cariño, constancia y sabiduría han sido un faro en mi vida, le doy las gracias por su constante apoyo y orientación. Su ejemplo de perseverancia y trabajo arduo han sido una inspiración para mí en cada paso del camino.

Al director Mateus, mi gratitud por su orientación experta y su compromiso con mi crecimiento académico es infinita. Sus consejos sabios y su paciencia han sido fundamentales en la realización de este trabajo.

Agradezco también a la codirectora Sindy, cuya orientación y apoyo han sido invaluable en cada etapa de este proceso, al igual que sus consejos e ideas me han presentado un mundo de oportunidades a explorar.

A todos mis amigos que han contribuido con sus conocimientos, su tiempo y su ánimo, les estoy profundamente agradecido. Este logro es el resultado del esfuerzo colectivo de muchos, incluyéndolos.

Para mis seres queridos, cuya fe en mí nunca vaciló, les doy las gracias por su amor incondicional y su constante aliento.

Que este logro sea un testimonio de la bondad de Dios y del amor y apoyo de quienes me rodean.

Autoridades de la Universidad

RECTOR GENERAL

R.P. FRAY ÁLVARO JOSÉ ARANGO RESTREPO, O.P.

VICERRECTOR ADMINISTRATIVO Y FINANCIERO GENERAL

R.P. FRAY HERNÁN YESID RIVERA ROBERTO, O.P.

VICERRECTOR ACADÉMICO GENERAL

R.P. FRAY MAURICIO ANTONIO CORTÉS GALLEGO, O.P.

SECRETARIO GENERAL

Dra. INGRID LORENA CAMPOS VARGAS

DECANO DIVISIÓN DE INGENIERÍAS

R.P. FRAY JAVIER ANTONIO HINCAPIÉ ARDILA, O.P.

SECRETARIA DE DIVISIÓN

E.C. LUZ PATRICIA ROCHA CAICEDO

DECANO FACULTAD DE INGENIERÍA ELECTRÓNICA

PHD. CARLOS ANDRÉS TORRES PINZÓN

Nota de aceptación

_____ Firma del tutor

_____ Firma del jurado

_____ Firma del jurado

BOGOTÁ D.C. _____ DE 2024

Advertencia

La Universidad Santo Tomás no se hace responsable de las opiniones y conceptos expresados en el trabajo de grado, solo velará por qué no se publique nada contrario al dogma ni a la moral católica y porque el trabajo no tenga ataques personales y únicamente se vea el anhelo de buscar la verdad científica.

Capítulo III –Art. 46 del Reglamento de la Universidad Santo Tomás.

Índice general

List of Figures	VIII
List of Tables	IX
1. Planteamiento del problema	1
1.1. Planteamiento	1
1.2. Pregunta problema	2
2. Justificación	3
3. Impacto social	4
4. Estado del arte	6
4.1. Visión Artificial	6
4.2. Segmentación semántica	8
4.3. Brazos robóticos	8
4.3.1. ROS (Robot Operating System)	9
5. Marco Teórico	10
5.1. ROS (Robot Operating System)	10
5.1.1. Image Based Visual Servoing (IBVS)	10
5.2. Python	11
5.3. Visión Artificial	11
5.3.1. Deep Learning	11
5.3.2. Redes de Convolución Neuronal (CNN)	11
5.3.3. Redes Neuronales Recurrentes (RNN)	11
5.3.4. Redes de Memoria de Largo y Corto Plazo (LSTM)	12
5.3.5. Modelo YOLO	12
5.4. Visión por Computadora	12
5.4.1. Detección de objetos	12
5.4.2. Segmentación Semántica	13
5.4.3. Segmentación de Instancias	13
5.5. Matrices de Rotación	13
6. Objetivos	14

6.1. Objetivo general	14
6.1.1. Objetivos específicos	14
6.2. Alcance	14
7. Metodología	16
7.1. Introducción	16
7.2. Fase 1 (Etapa de identificación)	16
7.2.1. Consulta bibliográfica	16
7.3. Fase 2 (Etapa de diseño)	16
7.3.1. Método	16
7.3.2. Recopilación de datos	17
7.3.3. Desarrollo del Sistema	17
7.3.4. Entrenar modelos	17
7.4. Fase 3 (Etapa de implementación)	17
7.5. Fase 4 (Etapa de evaluación)	17
7.5.1. Resultados	17
7.5.2. Costos y Beneficios	18
7.5.3. Conclusiones	18
8. Desarrollo Conceptual	19
8.1. Componentes	19
8.1.1. Cámara Realsense D435	19
8.2. Brazos Robóticos	20
8.2.1. Reactor X-200	21
8.2.2. Componentes ReactorX-200	22
8.2.2.1. Servomotores XM430-W350-T (3x)	22
8.2.2.2. Servomotores XL430-W250-T (4x)	22
8.2.2.3. Módulos	23
8.3. Configuración del Sistema	23
8.4. Visualización	24
8.5. Captura de Imágenes y Procesamiento de Datos	24
8.6. Transformación de Coordenadas	26
8.7. Control del Brazo Robótico	30
9. Resultados	31
9.1. Sistema de identificación	31
9.1.1. Entrenamiento del modelo YOLO	31
9.1.2. Evaluación del modelo YOLO	33
9.2. Sistema de recolección	38
10. Conclusiones y trabajos futuros	40
10.1. Conclusiones	40
10.2. Trabajos futuros	41
11. Presupuesto	42

12. Cronograma	43
-----------------------	-----------

Bibliografia	44
---------------------	-----------

Índice de figuras

1.	Cámara Realsense D435	20
2.	ReactorX-200	21
3.	XM430-W350-T	22
4.	XM430-W350-T	22
5.	Dynamixel U2D2	23
6.	Montaje de cámara Realsense sobre brazo	23
7.	(a) Enlaces del robot (b) Modelo del robot aplicado a los enlaces correspondientes	24
8.	Segmentación obtenida	25
9.	Posición para iniciar la detección	26
10.	Vista de cámara aplicada al plano del brazo en el programa RViz	27
11.	Caja comparada con la posición de un objeto en el plano real	28
12.	Resultados entrenamiento y validación	32
13.	Métricas obtenidas	32
14.	F1-Confidence	34
15.	Precision-Confidence	34
16.	Precision-Recall	35
17.	Matriz confusión	36
18.	Matriz normalizada	37
19.	Parámetros, coordenadas del objeto y coordenadas finales del brazo	38

Índice de cuadros

1.	Resultados de los Experimentos en la misma posición	39
2.	Resultados de los Experimentos variando la posición	39
3.	Presupuesto	42
4.	Cronograma	43

Capítulo 1

Planteamiento del problema

1.1. Planteamiento

Existe una disminución identificada en la cantidad de trabajadores en el área agrícola para el oficio de recolección debido a las generaciones más jóvenes están ignorando esta labor por falta de interés; esta situación ha generado la necesidad de adoptar nuevas tecnologías que permitan reemplazar la mano de obra faltante con herramientas para automatizar diferentes procesos en el campo [1] [2].

Esta disminución de mano de obra por parte de las generaciones más jóvenes en zonas rurales es debida a las distintas problemáticas económicas y sociales, generando diferencias entre el mundo urbano y rural, con brechas significativas que impactan en el desarrollo de las poblaciones rurales. Se percibe en las nuevas generaciones que el empleo urbano requiere menor esfuerzo físico, permite una mejor calidad de vida y brinda fácil acceso a servicios públicos [3].

Una situación que ejemplifica lo anterior es el estudio presentado en [4] en el que se destaca que "la actividad agropecuaria del Caribe colombiano (agricultura, ganadería, caza, silvicultura y pesca) redujo a la mitad su participación en el PIB regional, al pasar de representar el 14,9 % en 1997 al 7,3 % en 2016".

La situación del sector agrícola es desfavorable en estos momentos debido a problemáticas relacionadas con las inversiones hacia las zonas rurales; estas problemáticas afectan a la hora de presentar alternativas para mejorar la calidad de vida y reducir la carga laboral de los campesinos. Ejemplo de lo anterior son las bajas opciones de movilidad, que afecta enormemente al

transporte y desplazamiento de los productos como consecuencia del extremo deterioro en el que se encuentran las vías [5].

Dada la previa contextualización de las problemáticas en el campo colombiano, desde la desigualdad urbano-rural, la poca inversión en el sector agrícola y vías terciarias, los elevados costos de vivienda y la reducción de población trabajadora, se ha buscado una solución que abarca la robótica aplicada a brazos articulados, manipuladores, visión artificial y segmentación semántica para la clasificación de objetos. Para lograr lo anterior, se ha llegado a la pregunta problema presentada a continuación.

1.2. Pregunta problema

¿Cómo utilizar la segmentación semántica y el control servo-visual para desarrollar un sistema de recolección de frutas mediante el uso de un brazo robótico modelo ReactorX-200?

Capítulo 2

Justificación

El constante crecimiento e innovación de la tecnología en la actualidad, ha abierto la posibilidad de implementar nuevas alternativas de solución y aplicación de tecnologías de vanguardia en el sector agrícola. A pesar de esto, son pocas las iniciativas en este sentido y el campo colombiano sigue siendo una zona que no ha aprovechado al máximo este escenario caracterizado por los grandes avances en la Inteligencia Artificial y la robótica.

Este proyecto busca, con base en lo expuesto en el párrafo anterior, aportar en la identificación de alternativas tecnológicas y su aplicabilidad en un sector caracterizado por su baja tecnificación: según los resultados departamentales del tercer Censo Nacional Agropecuario, del total del área rural censada equivalente a 111.5 millones de hectáreas, solo el 6,4 % está destinada a cultivos (7.1 millones de hectáreas). De igual forma, el 15,9 % de las UPA (Unidades Productoras Agropecuarias) censadas afirmaron contar con algún tipo de maquinaria para el desarrollo de sus actividades agropecuarias, mientras que el 84,1 % de ellas declaró no poseer maquinaria [6].

Al optar por contribuir en la solución a esta problemática y teniendo en cuenta el método propuesto, se espera generar un impacto positivo no solo en la innovación e implementación de tecnologías en el área agrícola, si no que también se aborda la problemática de la falta de mano de obra al aportar, desde un punto de vista de desarrollo de ingeniería, mecanismos apropiados para soportar tareas específicas de recolección.

Capítulo 3

Impacto social

Los resultados de este proyecto se centran en contribuir a la producción de un método que aborda la recolección de alimentos en la agricultura, basado en la tarea de recolectar objetos mediante un brazo robótico utilizando segmentación semántica, simplificando el esfuerzo físico de la tarea de recolección.

Siguiendo el marco de la búsqueda de soluciones que promuevan el desarrollo sostenible a nivel global, esta investigación se enfoca en la recolección agrícola y la reducción de trabajo manual. Se alinea de manera directa con los Objetivos de Desarrollo Sostenible número 2 y 9 de las Naciones Unidas, teniendo definición: Poner fin al hambre, lograr la seguridad alimentaria y la mejora de la nutrición y promover la agricultura sostenible (ODS 2) y Construir infraestructuras resilientes, promover la industrialización inclusiva y sostenible y fomentar la innovación (ODS 9) [7]. En este contexto, el proyecto busca promover la agricultura sostenible (ODS 2) y promover la industrialización sostenible al igual que fomentar la innovación (ODS 9). La importancia de esta investigación radica en su potencial para aportar a un futuro más sostenible y equitativo, a través de los desafíos planteados por estos Objetivos de Desarrollo Sostenible.

El proyecto no busca dar una solución definitiva en la agricultura, sea nacional o mundial, ya que este proyecto pretende la investigación más no la implementación, los resultados servirán como base investigativa para aplicarse a futuro en un entorno agrícola. El motivo por el cual se plantea esta investigación utilizando visión artificial, es por su amplia documentación y anteriores aplicaciones las cuales coinciden en el uso de esta rama de la Inteligencia artificial, tanto para gestionar calidad de productos [8] como para identificación y conteo de estos[9] [10].

Esta investigación beneficiará al semillero de robótica de la Universidad Santo Tomás a través del desarrollo de visión artificial a la vez que refuerza el conocimiento sobre la manipulación

y simulación de un brazo robótico, y de su funcionamiento. Todo lo anterior sustentado desde las líneas e intereses de investigación del grupo GED.

Finalmente, este proyecto busca contribuir con la misión institucional de la Universidad Santo Tomás en relación a lo señalado en el documento Marco de Proyección Social en cuanto a la estrategia de desarrollo comunitario desde la construcción y promoción de las comunidades a través de la realización de proyectos de investigación, apoyo científico y técnico que surgen de la confluencia de intereses académicos y las dinámicas políticas, sociales, culturales y económicas de las comunidades. .

Capítulo 4

Estado del arte

El creciente desarrollo e integración de tecnologías y la robótica en la agricultura ha sido exponencialmente prometedor en cuanto a mejorar la eficiencia y la productividad en el sector. Se analizarán las investigaciones y desarrollos recientes en el ámbito de los brazos robóticos aplicados a la agricultura, destacando los avances en visión artificial y segmentación semántica así como la integración de ROS junto a Rviz y YOLO.

4.1. Visión Artificial

Es un área de la inteligencia artificial, que por medio de técnicas de procesamiento y segmentación de imágenes, así como el reconocimiento de estas, extrae mediante computadores digitales la composición y color para descifrar el entorno visualizado [11].

Tiene como objetivo deducir la estructura y propiedades de un entorno tridimensional a partir de una o de varias imágenes bidimensionales, recreando las propiedades geométricas del espacio en sí e incluso determinando propiedades de cada material y si hubo un cambio en este en medio de la detección [12].

El flujo de trabajo en el procesamiento de imágenes en cuanto a la visión artificial comprende las siguientes etapas:

- Adquisición de imagen: Se captura una imagen a través de una cámara para ser representada como una matriz de píxeles, donde cada píxel tiene una representación en color o intensidad mediante un valor dado.

-
- **Preprocesamiento:** Se prepara la imagen eliminando el ruido y mejorando el contraste, nitidez y demás aspectos para la correcta visualización de esta, para luego normalizar y ajustar el rango de píxeles y analizar posteriormente los resultados.
 - **Segmentación:** Se divide la imagen en segmentos o regiones donde cada segmento puede corresponder a un objeto o región de interés en la imagen. Algunas de las técnicas de segmentación incluyen:
 - **Segmentación basada en umbrales:** Segmentar la imagen en función de un umbral de valor de píxel.
 - **Segmentación basada en bordes:** Segmentar la imagen detectando los bordes entre diferentes regiones.
 - **Segmentación basada en regiones:** Segmentar la imagen agrupando píxeles con características similares.
 - **Representación de características:** Se extraen las características importantes obtenidas de la segmentación, ya sean descripciones matemáticas de la forma, textura o color de las regiones. Algunas de las técnicas en la representación de características incluyen:
 - **Histogramas:** Representan la distribución de valores de píxeles en una región.
 - **Descriptores de forma:** Representan la forma de una región, con datos como lo son el perímetro, área o momento central.
 - **Descriptores de textura:** Representar la textura de una región, como la rugosidad o la regularidad de su superficie.
 - **Clasificación o reconocimiento:** Se utilizan las características obtenidas para clasificar o reconocer el contenido de la imagen. Algunas de las técnicas de clasificación y reconocimiento comunes incluyen:
 - **Máquinas de vectores de soporte (SVM):** Un algoritmo de aprendizaje automático que puede usarse para clasificar datos en dos o más categorías.
 - **K-vecinos más cercanos (KNN):** Un algoritmo de aprendizaje automático que clasifica un nuevo punto de datos en función de la clase de sus vecinos más cercanos en el espacio de características.
 - **Redes neuronales artificiales (ANN):** Un modelo de aprendizaje automático inspirado en el cerebro humano que puede usarse para realizar tareas de clasificación y reconocimiento.

- Interpretación: Se interpreta el resultado de la clasificación o el reconocimiento, ya sea reconocer una pose, ubicación, una identidad o una acción.
- Salida: Se genera la salida del sistema de visión artificial, puede ser una descripción o clasificación, así como información relevante.

Proyectos tales como [13] [14] [15] se basan en la teoría del color (RVA o RGB) y reconocimiento del tamaño y forma de objetos, es decir, la geometría de estos. Coincidiendo con las propiedades de la visión artificial presentadas por [11], logrando presentar la importancia del procesado de imágenes con respecto al color y a la composición geométrica la cual es la base de la visión artificial.

4.2. Segmentación semántica

Siendo un conjunto de técnicas cuya función es asignar regiones dentro de una imagen para atribuir cierto significado semántico a cada una de estas, siendo una aplicación algorítmica de Deep Learning, esta tiene como propósito reconocer, clasificar y localizar objetos dentro de una o varias imágenes, siendo la segmentación el clasificar cada pixel de dicha imagen por instancias o categorías, para luego agrupar el segmento de pixeles que tengan algo en común en un recuadro (bounding boxes) y reconocer esto como un objeto [16].

4.3. Brazos robóticos

Los brazos robóticos o brazos manipuladores, son un conjunto de partes electromagnéticas que, mediante articulaciones, generan un sistema de movimiento a base de eslabones, estos dispositivos tienen distintos grados de libertad dando la posibilidad de, a mayor número de grados de libertad, tener mayores ángulos de movimiento y direcciones, permitiendo realizar tareas más complejas [17].

Estos elementos han sido un gran aporte a la robótica gracias a las distintas aplicaciones las cuales se pueden someter de manera eficaz, en aplicaciones industriales, médicas o de servicio público, juntándose con otros dispositivos para aumentar exponencialmente las posibilidades de aplicarse en dichos campos.

4.3.1. ROS (Robot Operating System)

Es un marco de trabajo diseñado para escribir software de robots, siendo una colección de distintos recursos tanto para desarrollar como para establecer una funcionalidad en un robot, teniendo como objetivo simplificar la creación de determinado comportamiento robótico, ya sea sencillo o complejo, para ser aplicado en distintas plataformas o software, lo cual no estaba presente antes de la implementación de este framework [18].

Ahora ROS es un estándar en la industria robótica, siendo el framework por defecto a la hora de desarrollar, implementar e innovar en las tecnologías de este sector, además de tener campo en el apartado industrial teniendo una versión dedicada a ser implementada en el sector, ganando notoriedad en la actualidad no solo por su enfoque robótico [19].

Capítulo 5

Marco Teórico

5.1. ROS (Robot Operating System)

El sistema ROS es una plataforma de código abierto muy utilizada en el campo de la robótica, que proporciona un entorno de desarrollo flexible y modular para el software de los robots, gracias a su arquitectura distribuida basada en nodos, ROS permite una comunicación eficiente entre los distintos componentes de un sistema robótico, lo que facilita la integración entre hardware y software. Debido a su versatilidad y a la amplia gama de herramientas y bibliotecas disponibles para la programación de robots, ROS se ha convertido en un estándar en la comunidad de robótica.

5.1.1. Image Based Visual Servoing (IBVS)

La servo-visión basada en imágenes (IBVS) es una técnica de control ampliamente utilizada en robótica que emplea información visual para dirigir el movimiento del robot al requerir de la necesidad de modelos precisos del entorno o del robot, proporciona una solución flexible y adaptable para llevar a cabo tareas de manipulación en entornos dinámicos y cambiantes, haciendo uso de las capacidades de procesamiento de imágenes de ROS y bibliotecas como OpenCV para calcular los comandos de control del robot basados en la retroalimentación visual, lo que posibilita la ejecución de tareas de manipulación con precisión y robustez.

5.2. Python

Python se ha convertido en el lenguaje de elección para la visión por computadora debido a su sintaxis clara, su amplia adopción en la comunidad de desarrollo y la disponibilidad de bibliotecas especializadas como OpenCV y TensorFlow. Estas bibliotecas proporcionan una variedad de herramientas y algoritmos para el procesamiento de imágenes y el aprendizaje automático, lo que permite a los desarrolladores implementar soluciones avanzadas en proyectos de visión por computadora de manera eficiente y efectiva.

5.3. Visión Artificial

Se enfoca en el desarrollo de sistemas para interpretar el contenido visual de imágenes y vídeos con el fin de generar resultados numéricos los cuales ser procesados y utilizados con otros fines.

5.3.1. Deep Learning

Tiene como función el entrenamiento de redes neuronales profundas para aprender representaciones de alto nivel de datos no estructurados con el fin de imitar las acciones del cerebro humano mediante neuronas, las cuales reciben y tratan información de anteriores neuronas para aumentar la precisión de los resultados.

5.3.2. Redes de Convolución Neuronal (CNN)

Son una clase de arquitecturas de aprendizaje profundo ampliamente utilizadas en tareas de visión por computadora. Estas redes están especialmente diseñadas para procesar datos de tipo matricial, como imágenes, y son capaces de aprender representaciones jerárquicas de características visuales.

5.3.3. Redes Neuronales Recurrentes (RNN)

Las redes neuronales recurrentes (RNN) son un tipo de arquitectura de redes neuronales diseñadas específicamente para modelar secuencias de datos, como series temporales o secuencias de texto. A diferencia de las redes neuronales convolucionales (CNN), que procesan datos en

paralelo, las RNN tienen conexiones retroalimentadas que les permiten mantener y actualizar estados internos a lo largo de la secuencia. Esto les otorga la capacidad de capturar dependencias temporales y contextuales en los datos de entrada.

5.3.4. Redes de Memoria de Largo y Corto Plazo (LSTM)

Las redes de memoria de largo y corto plazo (LSTM) son una extensión de las RNN diseñadas para abordar el problema de la pérdida de información a largo plazo, estas celdas de memoria están diseñadas para mantener y actualizar la información a lo largo del tiempo, lo que permite a las LSTM capturar dependencias a largo plazo en los datos de entrada, están equipadas con mecanismos de puerta que regulan el flujo de información dentro de la red.

5.3.5. Modelo YOLO

YOLO (You Only Look Once) es un modelo de detección de objetos en tiempo real con capacidad de realizar detecciones rápidas y precisas en imágenes, este opera dividiendo la imagen en una cuadrícula y prediciendo cajas delimitadoras y probabilidades de clase para cada celda de la cuadrícula, para luego aplicar "supresión no máxima", la cual elimina objetos duplicados dentro del mismo objeto y entrega un único objeto el cual es el más exacto.

5.4. Visión por Computadora

La visión por computadora es un campo interdisciplinario que se enfoca en desarrollar algoritmos y técnicas para permitir a las máquinas entender y analizar imágenes y videos digitales. Se enfoca en la automatización de tareas relacionadas con el procesamiento de imágenes, como la detección de objetos, la segmentación de imágenes y el reconocimiento de patrones, utilizando métodos basados en algoritmos y modelos matemáticos para extraer información significativa de las imágenes.

5.4.1. Detección de objetos

Es un proceso con el fin de identificar la presencia y ubicación de objetos específicos dentro de imágenes o videos, asignándoles una etiqueta de clase correspondiente, implica el uso de

algoritmos y técnicas de aprendizaje automático, como las redes neuronales convolucionales para reconocer y extraer características relevantes de los datos visuales proporcionados.

5.4.2. Segmentación Semántica

Es una técnica de visión por computadora que asigna una etiqueta semántica a cada píxel en una imagen. A diferencia de la detección de objetos, esta proporciona una comprensión detallada de la distribución espacial de diferentes clases de objetos en una imagen.

5.4.3. Segmentación de Instancias

Es una extensión de la segmentación semántica la cual, además de asignar una etiqueta semántica a cada píxel, distingue entre diferentes instancias de la misma clase de objeto, esto permite que cada objeto individual en una imagen se identifique y se distinga de otros objetos de la misma clase.

5.5. Matrices de Rotación

Permiten representar y controlar la orientación de objetos en un espacio tridimensional, estas matrices son cruciales para alinear y registrar imágenes, así como para calibrar sistemas de visión en entornos 3D, su función principal es describir la transformación de coordenadas de un objeto de un sistema de referencia a otro, lo cual es fundamental para llevar a cabo operaciones de procesamiento de imágenes y la toma de decisiones automatizadas.

Capítulo 6

Objetivos

6.1. Objetivo general

Desarrollar un sistema de visión por computadora basado en segmentación semántica para tareas de clasificación y recolección de objetos mediante el uso de un brazo robótico modelo ReactorX-200 y ROS.

6.1.1. Objetivos específicos

- Desarrollar un sistema de visión que integre las tecnologías de segmentación semántica mediante la librería TensorFlow y el framework ROS.
- Integrar Control por Realimentación Visual (Image Based Visual Servo, IBVS) con las imágenes obtenidas.
- Evaluar el sistema empleando métricas estándar tales como el porcentaje de acierto de la segmentación semántica y en el error de posición del IBVS en un entorno agrícola controlado.

6.2. Alcance

Este proyecto busca ser una primera aproximación del problema usando un entorno controlado, es decir, un entorno artificial, utilizando decoración como hojas y ramas además del uso diferentes intensidades lumínicas simulando la intensidad del sol a distintas horas del día,

además de una pequeña máquina de humo dando aspecto de un entorno con bruma. Contando con distintas imágenes para su posterior procesamiento utilizando TensorFlow para realizar la segmentación semántica, los recursos utilizados serán el espacio disponible en el laboratorio de robótica de la Universidad Santo Tomás, además de emplear el brazo robótico ReactorX200.

Capítulo 7

Metodología

7.1. Introducción

En la investigación se busca cumplir con las fases de desarrollo generales de un proyecto, las cuales constan de identificación, diseño, implementación y evaluación.

7.2. Fase 1 (Etapa de identificación)

7.2.1. Consulta bibliográfica

Consulta de documentación sobre ROS sobre el brazo robótico modelo ReactorX200, aplicaciones previas de segmentación semántica, robótica en la agricultura y control Servo-Visual basado en imágenes (IBVS), identificar tecnologías relevantes como las técnicas de procesamiento de imágenes y deep learning.

7.3. Fase 2 (Etapa de diseño)

7.3.1. Método

Se propone utilizar el marco de trabajo TensorFlow para la segmentación semántica, simulando el entorno mediante Gazebo y las librerías correspondientes al brazo robótico ReactorX200.

7.3.2. Recopilación de datos

Se capturan imágenes de un campo simulado utilizando una cámara montada en un brazo robótico, luego se anotan las imágenes para etiquetar objetos de interés y diferenciarlos del fondo.

7.3.3. Desarrollo del Sistema

Se desarrolla un sistema que integra un brazo robótico con una cámara para la adquisición de imágenes, luego se programa un algoritmo de segmentación semántica utilizando TensorFlow para ser integrada con el brazo robótico para clasificar y recolectar de objetos.

7.3.4. Entrenar modelos

Se procede a entrenar el modelo para segmentación semántica con los datos recopilados para ser ajustados junto con los parámetros aumentando la precisión.

7.4. Fase 3 (Etapa de implementación)

Se integra el sistema en un entorno agrícola simulado y se realizan pruebas para posteriormente evaluar la capacidad de clasificar y recolectar objetos del mismo, así como medir la eficiencia del algoritmo implementado.

7.5. Fase 4 (Etapa de evaluación)

7.5.1. Resultados

Se evalúan los resultados en términos de precisión en la clasificación y velocidad de recolección, para luego comprobar la eficacia del método propuesto.

7.5.2. Costos y Beneficios

Se analizan los costos para implementar un sistema físico en comparación a la mano de obra y el aumento de la productividad.

7.5.3. Conclusiones

Se anotan los hallazgos y en base a estos se hacen observaciones y recomendaciones para la implementación futura de sistemas similares.

Capítulo 8

Desarrollo Conceptual

El sistema se diseña en base a dos puntos clave, el primero es la identificación, la cual se realiza mediante un modelo de identificación de objetos y segmentación semántica en tiempo real a través del modelo YOLO; el segundo es la recolección, para la cual se propuso utilizar un brazo robótico ReactorX-200 para lograr esta tarea. El diseño tiene como fin el integrar estos puntos clave en un entorno controlado para obtener una solución con la capacidad de identificar y recolectar el objeto identificado.

8.1. Componentes

Se seleccionaron los siguientes componentes los cuales presentan características fundamentales para el desarrollo del proyecto: La cámara Realsense D435 debido a su tamaño y la capacidad de generar una imagen de profundidad la cual es compatible con la nube de puntos de RViz; y el brazo robótico ReactorX-200 por disponibilidad del semillero de robótica, el cual posee todas las características necesarias para ser utilizado junto a la cámara con el fin de cumplir con los objetivos propuestos.

8.1.1. Cámara Realsense D435

Una cámara diseñada para tareas de profundidad con un rango de hasta 10 metros, con dimensiones de 90 mm × 25 mm × 25 mm, y conexión mediante cable USB-C 3.1 Gen 1.

Cuenta con tecnología estereoscópica a una distancia mínima de aproximadamente 28 cm además de incluir un sensor de imagen de obturador global con campo de visión de $87^\circ \times 58^\circ$ con resolución de hasta 1280×720 con una tasa de cuadros de máximo 90 fps.

Para la captura de imágenes en color, ofrece una resolución 1920×1080 con una tasa de cuadros de 30 fps, con un sensor de imagen de obturador rodante con un campo de visión de $69^\circ \times 42^\circ$.



FIGURA 1: Cámara Realsense D435

8.2. Brazos Robóticos

Los brazos robóticos están caracterizados por su capacidad para realizar tareas de manipulación con precisión siendo esenciales en entornos industriales y de investigación avanzada, se componen de una serie de elementos mecánicos (como servomotores y engranajes), eléctricos y de software.

Los brazos robóticos son diseñados con una variedad de grados de libertad, lo que define la cantidad de movimientos independientes que pueden ejecutar, entre los cuales están los movimientos lineales o rotatorios, permitiendo al brazo posicionarse en una variedad de orientaciones y alcanzar objetivos en diferentes puntos de un espacio de trabajo.

La arquitectura típica de un brazo robótico se define por una estructura mecánica, siendo una serie de articulaciones conectadas por motores, y un sistema de control que coordina los motores y su velocidad, además de procesar la retroalimentación sensorial. Los motores, que pueden ser clasificados en motores eléctricos, hidráulicos o neumáticos, son responsables de generar el movimiento en las articulaciones del brazo, generalmente actuando en conjunto con información proporcionada por sensores, como la posición, velocidad y fuerza ejercida.



FIGURA 2: ReactorX-200

8.2.1. Reactor X-200

Es un brazo robótico modular altamente versátil y preciso diseñado para satisfacer las demandas de aplicaciones industriales y de investigación avanzada. Entre sus características destacadas se incluyen 5 grados de libertad, ofreciendo una amplia gama de movimientos y posiciones con un alcance de 550 mm y una envergadura de 1100 mm, esenciales para la tarea de recolección propuesta para este proyecto.

Ofrece una repetibilidad de 2.5 mm y una precisión de 5-8 mm, lo cual ofrece resultados precisos en tareas repetitivas, las cuales están inherentemente ligadas a la tarea de recolección.

Teniendo una capacidad de carga de máximo 150 g, se pueden manipular una variedad de objetos los cuales tendrán semejanza con algunas frutas pero teniendo un tamaño menor para garantizar una correcta recolección sin correr riesgo de fallas por una carga excesiva o falta de agarre.

8.2.2. Componentes ReactorX-200

8.2.2.1. Servomotores XM430-W350-T (3x)



FIGURA 3: XM430-W350-T

Estos servomotores utilizan un algoritmo de control PID para garantizar movimientos más suaves y estables, con una resolución de 4096 pulsos por revolución y un *ratio* de de 353.5:1, cuentan con una tasa de Baudios ajustable de 9,600 a 4.5 Mbps.

8.2.2.2. Servomotores XL430-W250-T (4x)



FIGURA 4: XM430-W350-T

Al igual que los servomotores XM430-W350-T, los servomotores XL430-W250-T emplean un control PID con una resolución de 4096 pulsos por revolución y una tasa de Baudios ajustable de 9,600 a 4.5 Mbps, teniendo como principal diferencia un *ratio* de 258.5:1 lo cual reduce el margen de torque a comparación de los otros servomotores.

8.2.2.3. Módulos

DYNAMIXEL U2D2

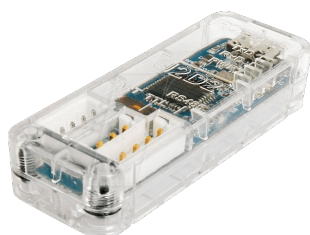


FIGURA 5: Dynamixel U2D2

Es un convertidor de comunicación USB de tamaño reducido que permite controlar y operar los servomotores DYNAMIXEL del ReactorX-200 desde una PC. Este módulo es compatible con conectores TTL de 3 pines y conectores RS-485 de 4 pines para conectar una variedad de servomotores DYNAMIXEL.

8.3. Configuración del Sistema

Describe cómo se ensamblan y configuran los componentes del sistema, incluida la cámara Realsense D435, el brazo robótico ReactorX-200 y cualquier otro hardware o software relevante.



FIGURA 6: Montaje de cámara Realsense sobre brazo

8.4. Visualización

Con el montaje preparado, se procede a visualizar el brazo en un entorno de simulación, se utilizó la interfaz visual de ROS (RViz), con la cual se identifican los distintos enlaces presentes en el robot, los cuales indican las articulaciones y la posición en un plano XYZ de estas, así como la rotación de estas.

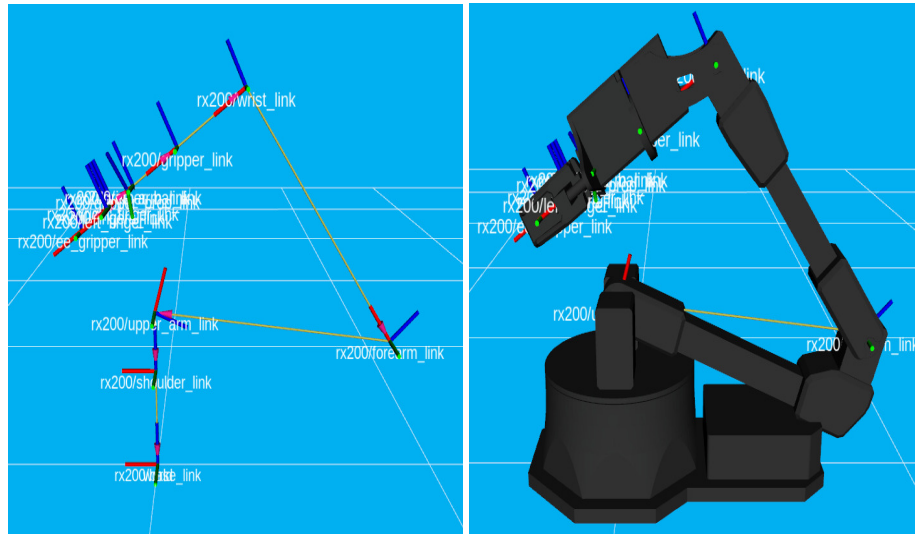


FIGURA 7: (a) Enlaces del robot (b) Modelo del robot aplicado a los enlaces correspondientes

8.5. Captura de Imágenes y Procesamiento de Datos

Para esto se utiliza un archivo con las configuraciones iniciales de la cámara, guardando esta configuración en una variable nombrada "dc" donde se inicializan, para luego llamar a una función que retorna los *frames* de RGB y de profundidad de la cámara.

```
dc = DepthCamera()
```

```
ret, depth_frame, color_frame = dc.get_frame()
```

Después de inicializar la cámara, se carga un modelo diseñado para segmentación y entrenado para reconocer distintas frutas, esto mediante la librería de YOLO, cabe destacar que se utilizó la mejor ponderación de este modelo entrenado, con el fin de obtener resultados más satisfactorios.

```
model = YOLO("best.pt")
```

En esta ponderación se incluyen los nombres de las clases pertenecientes al modelo entrenado las cuales provienen de su archivo de configuración, que en este caso son las distintas frutas presentes en las imágenes de entrenamiento y validación, haciendo uso de estos, se crea una variable la cual contiene estas clases:

```
classNames = ["Apple", "Banana", "Grapes", "Kiwi", "Mango", "Orange", "Pineapple",  
             "Strawberry", "Sugarapple", "Watermelon"]
```

Al finalizar la configuración de la cámara y del modelo, se realiza la predicción de los objetos mediante el *frame* de RGB para obtener los resultados los cuales serán impresos en una variable llamada "anotaciones".

```
resultados = model.predict(color_frame, imgsz = 640)
```

```
anotaciones = resultados[0].plot()
```

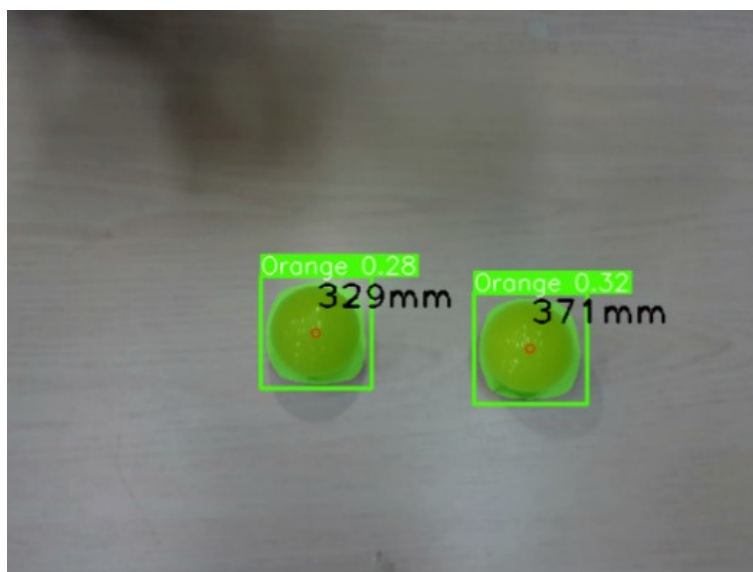


FIGURA 8: Segmentación obtenida

Estos resultados anotados contienen parámetros los cuales son encontrados en las cajas de los objetos, de los cuales nos interesan específicamente la posición de la caja en coordenadas XY al igual que el alto y ancho del objeto contenidos en el tensor.

```
dim = box.xywh
```

```
coord = dim.cpu().numpy()
```

```
x, y, width, height = coord[0]
```

Se genera un punto central en la caja el cual representa el centro del objeto.

```
point = (int(x), int(y))
```

8.6. Transformación de Coordenadas

Obtenidos los parámetros necesarios mediante YOLO, se necesita conocer el lugar en el cual está la cámara con respecto al brazo para transformar las coordenadas 2D del objeto a un plano 3D, para esto se debe tener en cuenta la pose en la cual el robot inicia la detección de objetos, la cual fue establecida en $(0.1, 0, 0.3)$ con un *pitch* de 1.2



FIGURA 9: Posición para iniciar la detección

Conociendo la pose del brazo, se realizó la tarea manual en RViz de publicar la transformada de la cámara hasta encontrar su respectiva posición y orientación, utilizando el siguiente comando:

```
roslaunch static_transform_publisher
```

Con los elementos:

- X
- Y
- Z
- *yaw*
- *pitch*
- *Roll*
- *Frame* Padre
- *Frame* Hijo
- Periodo en milisegundos

Después de introducir estos parámetros, se crea un enlace en RViz el cual representa virtualmente la posición de la cámara, más específicamente de los *frames*, por lo cual este comando se tuvo que ejecutar con los *frames* de RGB, profundidad y ópticos, al igual que el enlace de la cámara.

Para verificar la correcta posición y rotación de la cámara, se utiliza la herramienta de PointCloud2 para renderizar y visualizar la nube de puntos generada por cada *frame* de esta, concordando con el montaje en el plano físico.

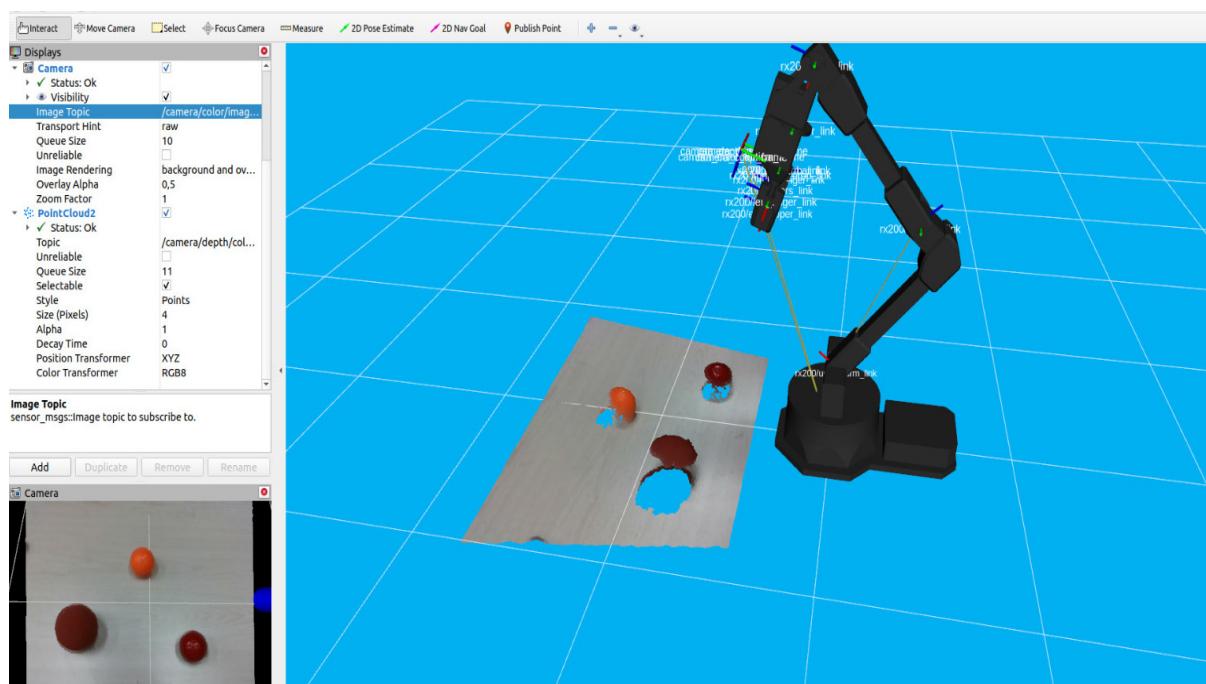


FIGURA 10: Vista de cámara aplicada al plano del brazo en el programa RViz

Dando como resultado estas coordenadas con su respectiva rotación:

$camera_position = np.array([0,125, 0, 0,36])$ $camera_orientation = np.array([90, 180, 20])$

Por medio de una simulación manual se ubica un objeto en una posición determinada, se genera una caja virtual mediante el uso de la herramienta MotionPlanning en su apartado de Scene Objects, esto mide las coordenadas XYZ aproximadas del objeto sin necesidad de hacer cálculos, comprendiendo cómo funcionan las coordenadas en un plano tridimensional:

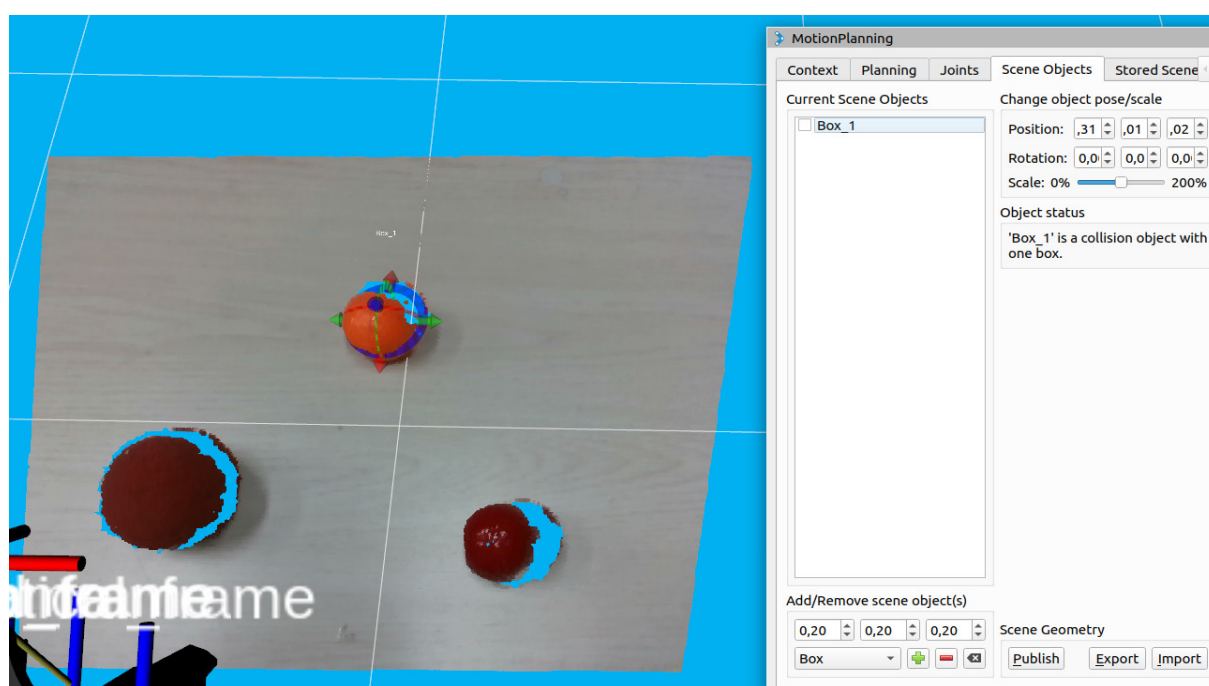


FIGURA 11: Caja comparada con la posición de un objeto en el plano real

Gracias a esta prueba se identifican los ejes principales y se denotan de manera: X (Mover el brazo adelante/atrás), Y (Mover el brazo hacia la derecha o izquierda) y Z (Elevar o bajar el brazo) así como las distancias que recorre el brazo con cada modificación.

Teniendo claridad sobre el plano 3D, se construye la matriz de rotación tridimensional compuesta a partir de las rotaciones individuales alrededor de cada eje previamente establecidas, esta matriz se compone de tres matrices individuales, una para cada eje de rotación (X Y Z), representadas como R_x , R_y , y R_z , de esta manera:

Rx:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_1 & -\sin \theta_1 \\ 0 & \sin \theta_1 & \cos \theta_1 \end{bmatrix} \quad (8.1)$$

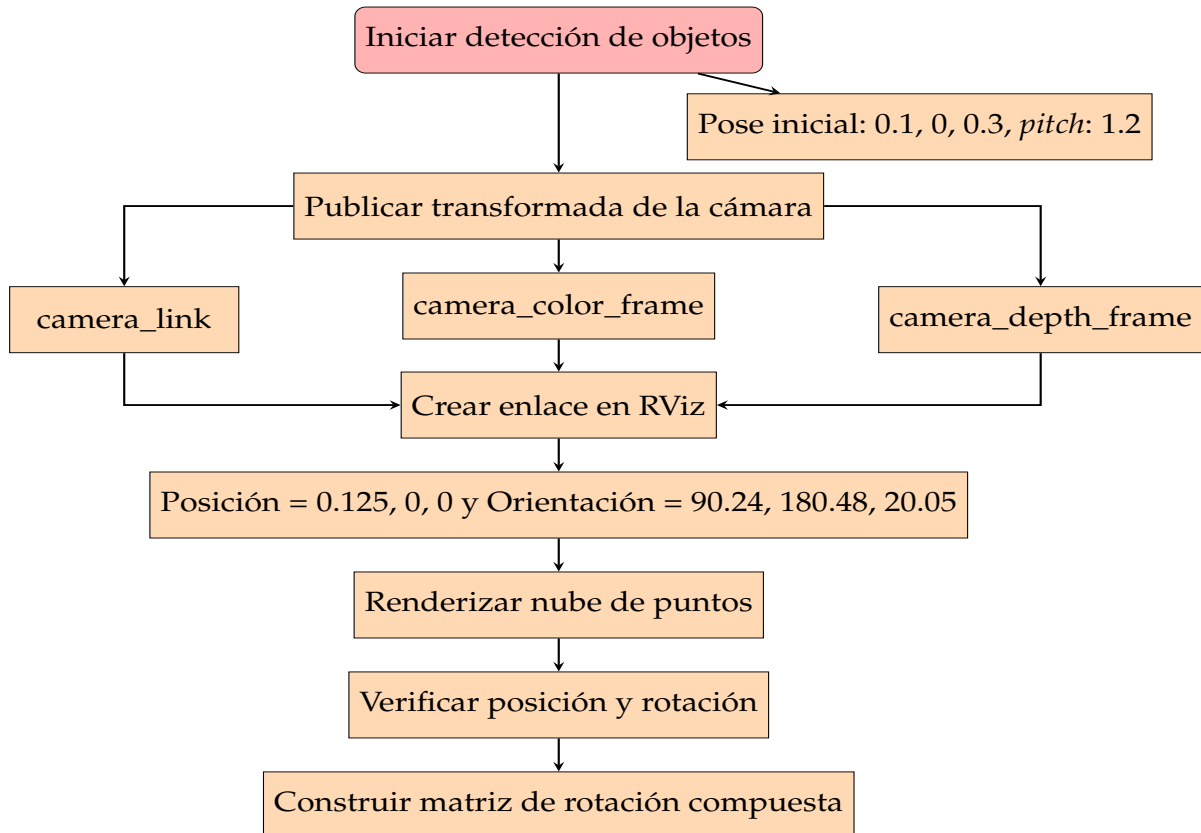
Ry:

$$\begin{bmatrix} \cos \theta_2 & 0 & \sin \theta_2 \\ 0 & 1 & 0 \\ -\sin \theta_2 & 0 & \cos \theta_2 \end{bmatrix} \quad (8.2)$$

Rz:

$$\begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 \\ \sin \theta_3 & \cos \theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8.3)$$

A continuación se presenta un diagrama de flujo que indica el proceso general para la construcción de la matriz de rotación:



Con las matrices para cada uno de los ejes, por medio de la función *np.dot* se realiza un producto de las tres matrices de rotación en el orden ZYX, primero aplicando la rotación en el eje X, luego en el eje Y, finalmente alrededor del eje Z

```
rotation_matrix = np.dot(Rz, np.dot(Ry, Rx))
```

Después de hacer ajustes con las coordenadas de la cámara tomando en cuenta la profundidad de la mesa con respecto a la cámara y añadiendo esto a la matriz rotacional, se obtienen las coordenadas a las cuales el brazo tendrá que ir.

8.7. Control del Brazo Robótico

Obteniendo las coordenadas clave para el movimiento del brazo, se deben importar las librerías de los manipuladores de la serie X de interbotix, las cuales proveen distintas funciones relacionadas al movimiento para facilitar el control:

```
from interbotix_xs_modules.arm import InterbotixManipulatorXS
```

Luego, se inicializa el modelo correspondiente a nuestro brazo, el cual carga la configuración además de identificar el número de grados de libertad, rotación y distancia máximas y límites de funcionamiento:

```
bot = InterbotixManipulatorXS("rx200")
```

Con el modelo del brazo inicializado y la librería cargada, se utiliza la función *bot.arm.set_ee_pose_coordinates* que indica al robot hacia qué posición ir, introduciendo valores de *x*, *y*, *z*, *roll*, *pitch* y *pitch*, los cuales indican la ubicación y la rotación que debe asumir el robot.

Además, se utilizan las funciones *bot.gripper.open()* y *bot.gripper.close()* después de cada movimiento para manipular el agarre del robot en función de lo que se requiere.

Capítulo 9

Resultados

9.1. Sistema de identificación

En esta sección, se presentan los resultados obtenidos durante el desarrollo e implementación del sistema de identificación y recolección donde se detallan los resultados del entrenamiento del modelo YOLO, al igual que la evaluación de su desempeño detectando los objetos de interés. Se evaluará la integración de estos resultados con el brazo robótico, destacando la precisión y la viabilidad del sistema en entornos controlados.

9.1.1. Entrenamiento del modelo YOLO

Es fundamental para el éxito del sistema propuesto el entrenar un modelo de segmentación, ya que es de gran importancia que el algoritmo tenga la capacidad de identificar y localizar objetos de interés a través de imágenes. Se llevó a cabo un entrenamiento de 100 épocas utilizando conjuntos de datos específicamente recopilados y anotados para identificar y segmentar frutas, dando como resultado un conjunto de gráficas correspondientes al entrenamiento/validación y un conjunto de gráficas con distintas métricas:

Se presenta el conjunto de gráficas de *train* y *val* con respecto a distintos aspectos como pérdidas de clases y segmentos, además de otras pérdidas:

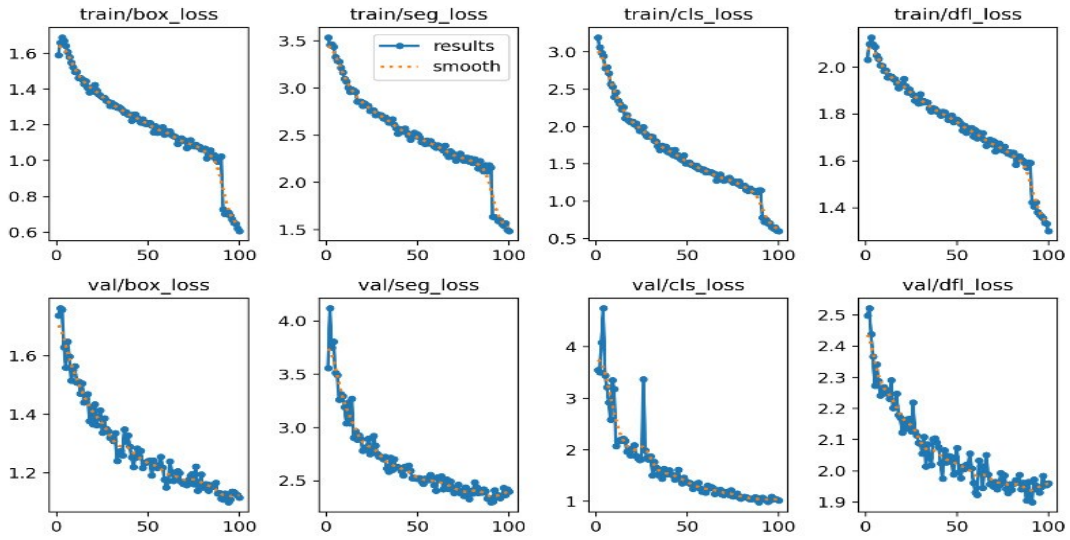


FIGURA 12: Resultados entrenamiento y validación

Las pérdidas de ambos grupos de gráficas son satisfactorios, a pesar de que hubo un súbito decremento de pérdidas cerca del final del entrenamiento y de la validación, los resultados generales presentan un comportamiento acorde frente a las gráficas generales de estos modelos, los cuales tienden a disminuir hasta llegar al máximo de épocas.

En la siguiente imagen se presentan las métricas resultantes del entrenamiento, donde se pueden evidenciar tanto la precisión como la precisión media promedio (mAP) además del *Recall*, las cuales son medidas que representan el rendimiento general del modelo, de esta manera:

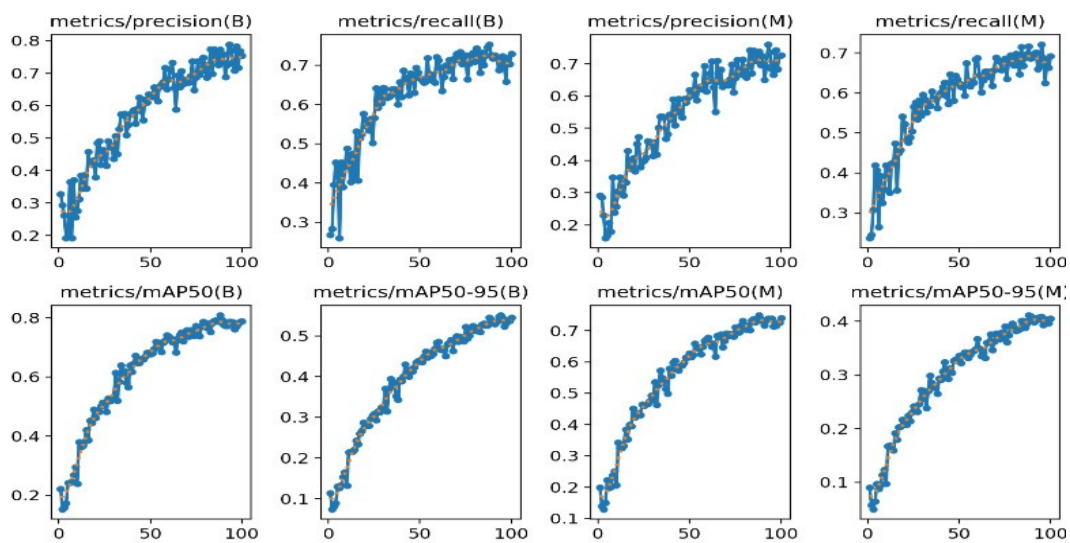


FIGURA 13: Métricas obtenidas

- **Precisión:** Indica cuántas detecciones fueron correctas.
- **mPA50:** Calculada con un umbral de intersección sobre unión (IoU) de 0,50, considera sólo las detecciones sencillas.
- **mAP50-95:** Calculada con distintos umbrales de IoU que van de 0,50 a 0,95. Para distintos niveles de dificultad en la detección.
- **Recall:** Calcula la proporción de verdaderos positivos entre todos los positivos reales, midiendo la capacidad del modelo para detectar todos los casos de una clase.

9.1.2. Evaluación del modelo YOLO

Es crucial evaluar el modelo YOLO para determinar la capacidad con la que detecta con precisión los objetos en diversas condiciones y entornos, se presentan los resultados de las pruebas de evaluación realizadas, que incluyen gráficas de precisión, sensibilidad y puntajes de F1 para después ser analizadas brevemente teniendo en cuenta los rangos en los que hay mayor desempeño.

- **Curva de puntuación F1:** Ofrece información sobre el equilibrio del modelo entre falsos positivos y falsos negativos en diferentes umbrales.
- **Curva Precisión-Recall:** La comparación de estas características representa a eficacia del modelo frente a clases desequilibradas.
- **Curva de precisión:** Ayuda a comprender cómo varía la precisión al cambiar el umbral.
- **Curva de Recall:** Ilustra cómo cambian los valores de recuerdo según los distintos umbrales.
- **Matriz de confusión:** Muestra los recuentos de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos de cada clase.
- **Matriz de confusión normalizada:** Simplifica la comparación del rendimiento entre clases.

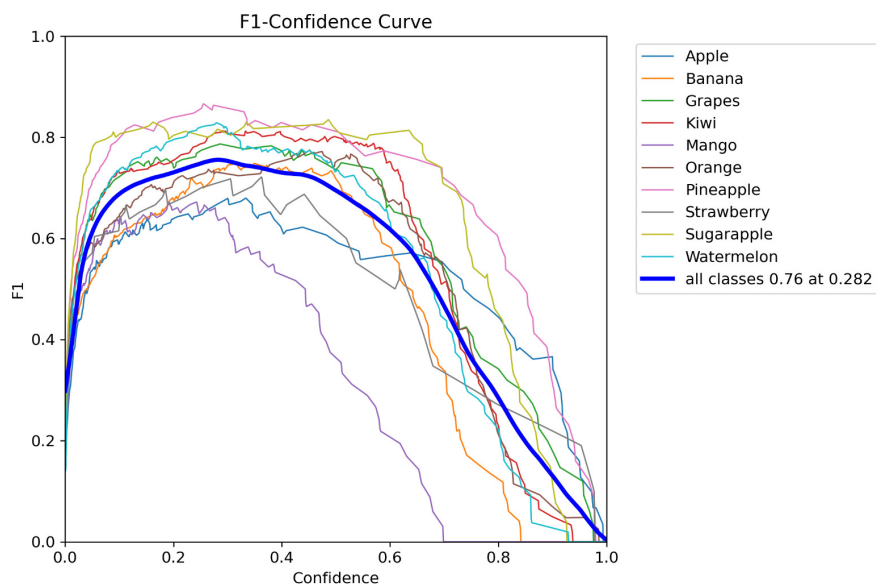


FIGURA 14: F1-Confidence

La figura 14 nos indica el equilibrio entre precisión y *recall* a medida que se ajusta el umbral de confianza, ayudando a determinar el umbral óptimo para las predicciones del modelo, siendo un valor ideal de F1 el más cercano posible a 1, se puede determinar que los valores recomendables de confianza van desde 0.2 hasta 0.4, mientras que los valores aceptables varían después de 0.4 hasta aproximadamente 0.7.

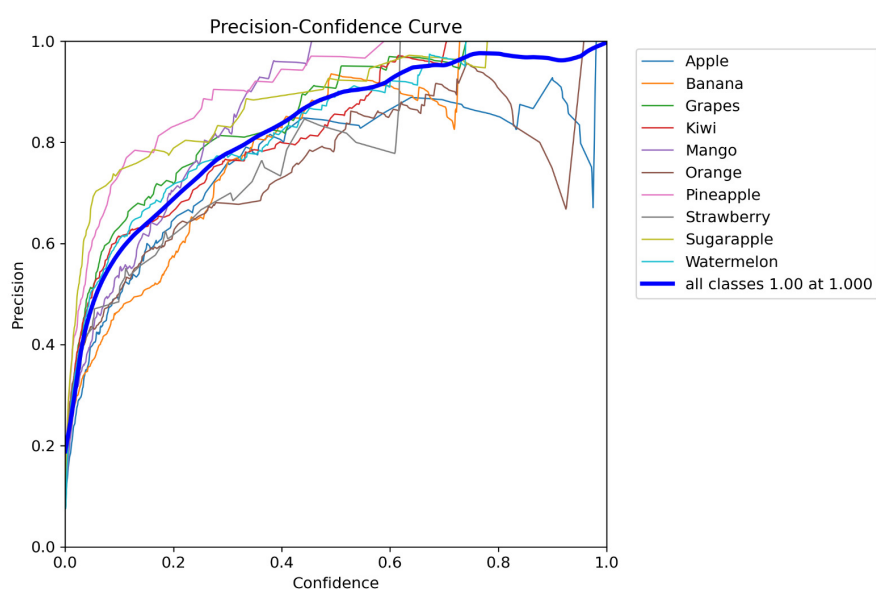


FIGURA 15: Precision-Confidence

La figura 15 permite visualizar los cambios de la precisión del modelo a medida que se modifica la confianza de sus predicciones, lo que ayuda a seleccionar un umbral de confianza adecuado para obtener un equilibrio entre precisión y el número de falsos positivos, presentando valores satisfactorios frente a una confianza por mayor de 0.5.

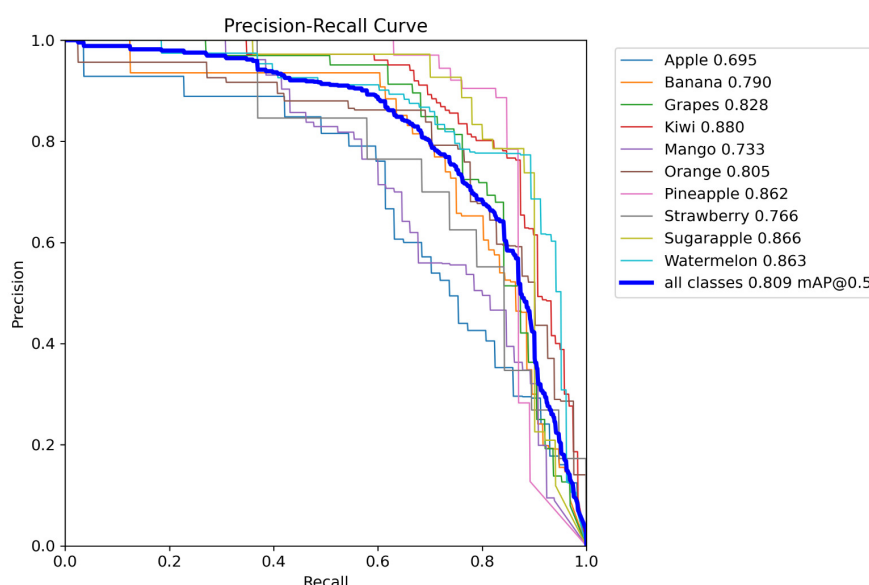


FIGURA 16: Precision-Recall

La figura 16 representa el cambio de la precisión en función del *recall* a medida que se ajusta el umbral de confianza, ofreciendo información sobre el rendimiento del modelo en un amplio rango de umbrales, habiendo mayor equilibrio entre los valores de 0.7 a 0.8.

Junto a las gráficas de entrenamiento y validación así como las métricas proporcionadas, están las matrices de confusión y confusión normalizada, las cuales nos proporcionan una visión detallada de la distribución de las predicciones del modelo en relación con las clases de los objetos detectados.

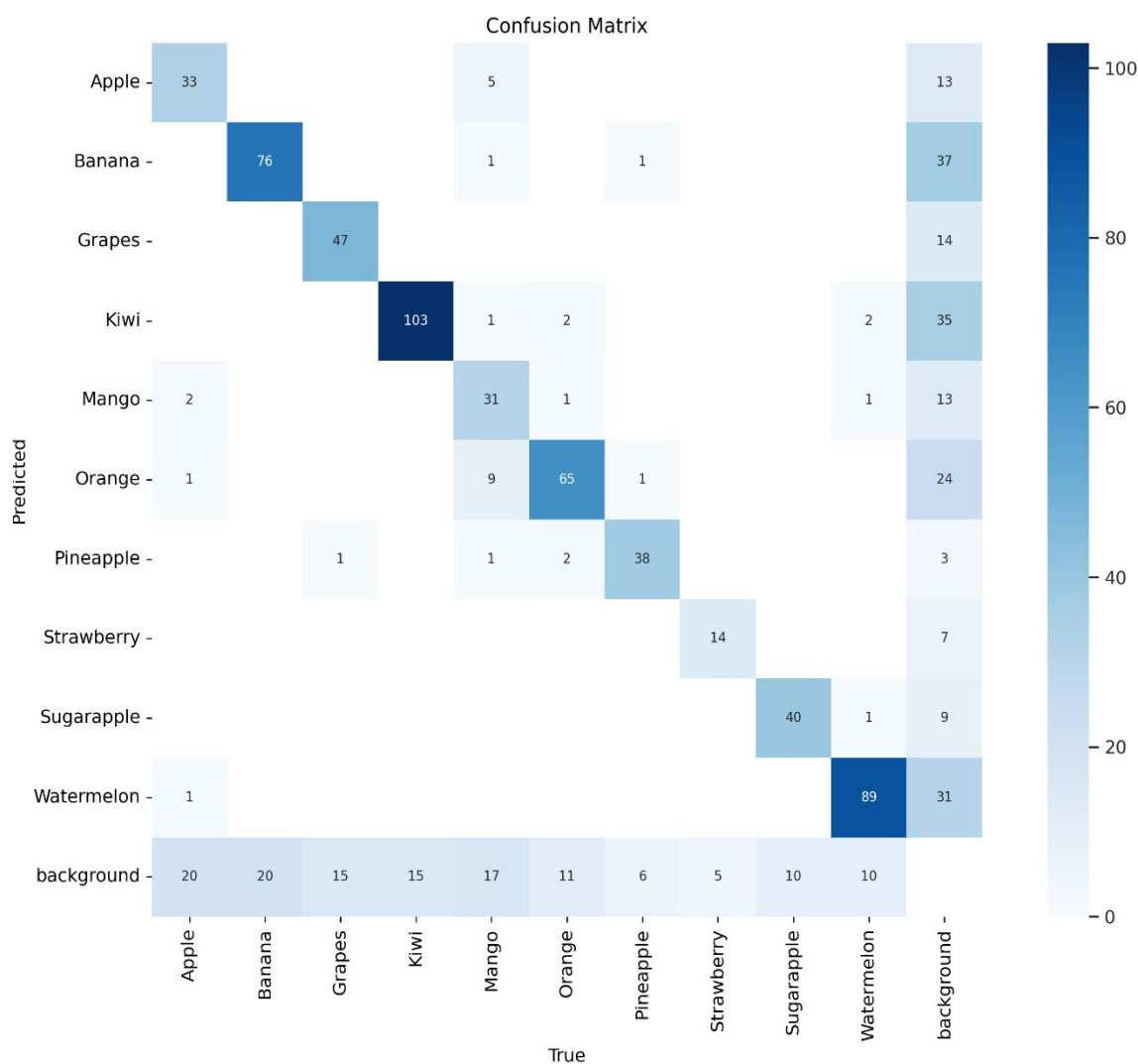


FIGURA 17: Matriz confusión

En la figura 17, es posible identificar dónde el modelo está detectando correctamente las clases, al igual que las situaciones donde presenta fallos, se puede observar si hay clases que el modelo tiende a confundir entre sí, lo que puede indicar similitudes visuales entre los objetos, por ejemplo, los mangos con las naranjas o las manzanas con los mangos.

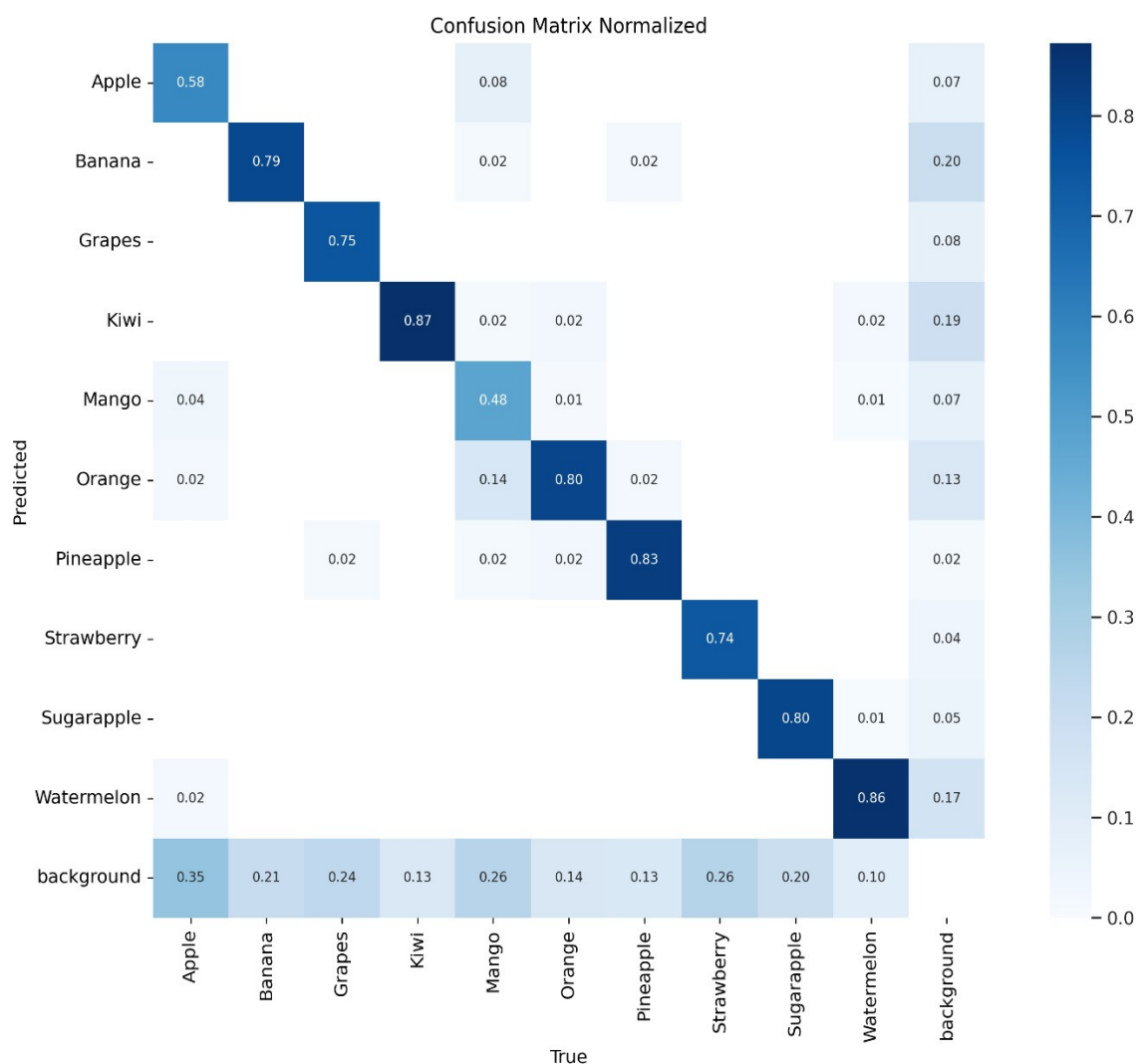


FIGURA 18: Matriz normalizada

En la figura 18 se presenta esta matriz normalizada, la cual es una versión en la que los valores de cada celda se dividen por el número total de muestras de la clase correspondiente, permitiendo comparar directamente la proporción de predicciones tanto correctas como incorrectas para cada clase, esto facilita la identificación de ciertos patrones de error entre las clases al igual que proporciona una vista de la precisión por clases del modelo.

9.2. Sistema de recolección

La fase de implementación del sistema de recolección se detalla en esta sección donde se recopilan los experimentos propuestos para evaluar la integración y operación del brazo robótico ReactorX200 en conjunto con el sistema de identificación, se analizan los resultados obtenidos durante las pruebas destacando la precisión y eficiencia de la segmentación al igual que la precisión de agarre.

Previamente en el capítulo de Desarrollo Conceptual, se procesaron los datos de manera que por medio de las cajas, se obtiene la ubicación de un objeto en un plano 2D (plano de la cámara), la cual es representada mediante un punto ubicado en el centro de este; y por medio de una transformación de coordenadas y matrices de rotación, se encuentra la ubicación del objeto en el plano XYZ el cual se imprime en consola junto a todos los parámetros extraídos de la caja, de esta forma:

```
0: 480x640 2 Oranges, 22.9ms
Speed: 1.5ms preprocess, 22.9ms inference, 1.7ms postprocess per image at shape (1, 3, 480, 640)
Este objeto es: Orange
Las coordenadas en formato xywh son: [[ 440.8 282.42 94.686 92.466]]
Shape de coord: (1, 4)
Valor de coord: [[ 440.8 282.42 94.686 92.466]]
Dimensiones del punto: (440, 282)
Coordenadas x del objeto: '440'
Coordenadas y del objeto: '282'
Coordenadas w del objeto: '94'
Coordenadas h del objeto: '92'
Coordenadas z del objeto: '371'
COORDENADAS DEL ROBOT:[ 0.30359 -0.064966 0.45573]
Este objeto es: Orange
Las coordenadas en formato xywh son: [[ 260.99 269.38 93.876 93.005]]
Shape de coord: (1, 4)
Valor de coord: [[ 260.99 269.38 93.876 93.005]]
Dimensiones del punto: (260, 269)
Coordenadas x del objeto: '260'
Coordenadas y del objeto: '269'
Coordenadas w del objeto: '93'
Coordenadas h del objeto: '93'
Coordenadas z del objeto: '330'
COORDENADAS DEL ROBOT:[ 0.32784 0.0025532 0.3773]
DETECTION:1
```

FIGURA 19: Parámetros, coordenadas del objeto y coordenadas finales del brazo

Estos resultados presentados tienen como objetos dos pelotas de *ping pong* naranjas con el suficiente parentesco a una naranja para ser reconocidas con una baja probabilidad, esto debido a que los dedos del robot presentan un área de agarre pequeña, por lo cual los objetos de este tamaño son adecuados para realizar la prueba debido a la falta de espacio en el agarre.

Estas pelotas, ubicadas en una mesa plana con un apoyo debajo de ellas para evitar movimientos no deseados, se encuentran respectivamente en (440,282) y (260,269) pixeles, generando coordenadas de (0.36, -0.06) y (0.32, 0.002).

Se realizaron 5 pruebas realizando el mismo experimento para evaluar características como repetibilidad y variaciones a la hora de la creación de coordenadas, teniendo ligeros cambios en la iluminación, resultando en:

CUADRO 1: Resultados de los Experimentos en la misma posición

N° Experimento	X1	Y1	Acierto	Error	X2	Y2	Acierto	Error
1	0.36	-0.06	X		0.32	0.002	X	
2	0.35	-0.07		X	0.32	0.003	X	
3	0.36	-0.07	X		0.33	0.003		X
4	0.34	-0.06		X	0.32	0.003		X
5	0.35	-0.06		X	0.32	0.002	X	

Al revisar el cuadro 1, se evidencian ciertos errores en la transformación de coordenadas, posiblemente debido a las variaciones en la iluminación, lo cual presenta cambios en las coordenadas en el plano 2D al no detectar enteramente la pelota como una fruta, resultando en las variaciones a la hora de transformar las coordenadas al plano 3D.

Se encontraron errores de posición en 3 objetos distintos durante esta prueba, es decir, un porcentaje de error del 30 % de los casos, los cuales pueden ser atribuidos a errores de calibración de cámara y el torque del motor de la base del robot.

Además de estas pruebas, se realizaron 5 pruebas adicionales, cada una con posiciones distintas para evaluar la transformación de coordenadas, dando como resultado:

CUADRO 2: Resultados de los Experimentos variando la posición

N	X1_p	Y1_p	X1_t	Y1_t	A	E	X2_p	Y2_p	X2_t	Y2_t	A	E
1	551	242	0.28	-0.07	X		253	261	0.32	0.004		X
2	260	244	0.32	0.008	X		322	245	0.31	-0.09	X	
3	414	266	0.30	-0.04		X	432	221	0.30	-0.032		X
4	110	227	0.34	0.06	X		167	234	0.33	0.04	X	
5	211	300	0.33	-0.002		X	125	245	0.32	0.07	X	

Al tener en cuenta los resultados del cuadro 2, se determinó que mientras el objeto esté cercano al centro de la imagen con respecto al eje Y (240 pixeles aproximadamente), aumenta la tasa de éxito al evaluar el sistema, al salirse del rango intermedio del eje Y, se presentan irregularidades a medida que aumenta o disminuye fuera del rango, de manera exponencial.

Capítulo 10

Conclusiones y trabajos futuros

10.1. Conclusiones

El proyecto investigativo abordó la integración de dos tecnologías, la segmentación semántica y el uso de brazos robóticos, permitiendo desarrollar un sistema de visión con el objetivo de controlar el dispositivo mediante coordenadas tridimensionales a partir de coordenadas bidimensionales proporcionadas por la cámara.

Se llevaron a cabo experimentos bajo un entorno controlado para después analizar el índice de aciertos de cada experimento, estas pruebas estaban sujetas a cambios en la posición de los objetos para determinar la precisión del sistema en general.

El sistema ha demostrado ser efectivo en la identificación de productos agrícolas con cambios en la iluminación no muy extremos, durante el desarrollo del proyecto se evidenciaron errores ocasionales al detectar prendas o parte de ellas como frutas, por lo cual se recomienda entrenar el modelo bajo otros parámetros para mejorar la curva de pérdidas y reducir estos fallos.

La eficiencia del sistema en términos de tiempo de recolección y movimientos del brazo robótico ha sido satisfactoria, con tiempos promedio de recolección en el rango de 7 a 10 segundos por objeto, con precisión en la recolección de alrededor del 60 % en el total de experimentos, se infieren varios puntos a mejorar tanto en el sistema de transformación de coordenadas como en mantenimiento del brazo, ya que presenta una deficiencia en el torque del motor de la base, teniendo que generar manualmente

el torque inicial ya que el motor no inicia con la fuerza necesaria para realizarlo, se propone mantenimiento general del dispositivo.

Se ha identificado la necesidad de mejorar la robustez del sistema en condiciones adversas, como cambios fuertes en la iluminación o la presencia de objetos invasivos alrededor de la imagen para realizar pruebas en distintos ángulos y con distintos objetos de fondo además de frutas.

10.2. Trabajos futuros

Se propone la investigación y desarrollo de técnicas avanzadas de segmentación semántica para mejorar la precisión en la identificación de los objetos propuestos, especialmente en situaciones adversas de iluminación y cambios de fondo.

Integrar técnicas de aprendizaje activo para permitir que el sistema mejore su rendimiento con el tiempo, mediante la retroalimentación continua de los usuarios.

Explorar la posibilidad de utilizar técnicas de aprendizaje por refuerzo para optimizar los movimientos del brazo robótico y reducir aún más los tiempos de recolección, además de un control PID para tener un movimiento más preciso y suave.

Ampliar el alcance del sistema para incluir la clasificación y recolección de una gama más amplia de productos agrícolas, también abarcando la posibilidad de detectar enfermedades o variaciones en las formas de una fruta.

Realizar pruebas adicionales en entornos agrícolas reales para validar la efectividad y robustez del sistema en condiciones del mundo real.

Diseñar un algoritmo de detección de agarre mediante procesamiento de imágenes y random forest, con el fin de ampliar los usos posibles al sistema en caso de ser utilizado con fines no agrícolas, como por ejemplo un robot de asistencia o de limpieza.

Integrar un sistema móvil para permitir un desplazamiento autónomo o controlado para resolver la problemática de ser un sistema estático, además de diseñar e imprimir una mano con más espacio en el agarre, para ampliar el área mínima en el que los objetos puedan ser recolectados.

Capítulo 11

Presupuesto

El presupuesto está estimado en base a las horas totales destinadas de cada miembro participe en el proyecto. Por parte de la institución se incluyen servicios tales como el servicio de internet, las licencias del software a utilizar y el acceso a recursos bibliográficos por medio del CRAI. El estudiante prestará su equipo para esta investigación.

Material	Cantidad	Unidad	Valor	Total	Responsable
Recursos Humanos					
Director	45	Horas	\$ 45.000	\$ 2.025.000	Universidad
Codirector	45	Horas	\$ 45.000	\$ 2.025.000	Universidad
Investigador	600	Horas	\$ 19.000	\$ 11.400.000	Estudiante
Suma Total				\$ 15.450.000	
Servicios y Material					
Computador	1	Unidad	\$ 1.800.000	\$ 1.800.000	Estudiante
Python	1	Licencia	\$ 0,00	\$ 0	Libre
Salón Robótica	1	Unidad	\$ 0,00	\$ 0	Universidad
Internet	600	Horas	\$ 1.000	\$ 600.000	Universidad
CRAI	30	Horas	\$ 54.000	\$ 1.620.000	Universidad
Suma Total				\$ 4.020.000	
Subtotal				\$ 19.470.000	
Imprevistos 10%				\$ 1.947.000	
Total				\$ 21.417.000	

CUADRO 3: Presupuesto

Capítulo 12

Cronograma

A continuación se presenta cronología del proyecto, teniendo como base la metodología previamente presentada, dividiendo la investigación en sus respectivas fases, con una planificación estimada de 6 meses, para un total de 22 semanas.

Etapa	Actividades	Duración Estimada (semanas)
Fase 1 (Identificación)	Revisión bibliográfica.	2
	Identificación de tecnologías relevantes.	
Fase 2 (Diseño)	Aprendizaje del marco de trabajo TensorFlow y herramienta de simulación Gazebo junto con ROS.	5
	Recopilación de datos (Captura de imágenes) y anotación de imágenes para etiquetar objetos de interés.	5
	Desarrollo del sistema (Integración de brazo robótico, cámara y programación de algoritmo).	
	Entrenamiento de modelos para segmentación semántica.	3
Fase 3 (Implementación)	Integración del sistema en entorno agrícola simulado.	4
	Pruebas y evaluación.	
Fase 4 (Evaluación)	Evaluación de resultados (Precisión y velocidad).	3
	Análisis de costos y beneficios.	
	Conclusiones y recomendaciones.	

CUADRO 4: Cronograma

Bibliografía

- [1] E. Mora Martínez et al., «Disminución de mano de obra joven en zonas rurales de la región caribe colombiana y su impacto en el desarrollo agrícola.», 2021.
- [2] W. Geza, M. Ngidi, T. Ojo, A. A. Adetoro, R. Slotow y T. Mabhaudhi, «Youth participation in agriculture: a scoping review», *Sustainability*, vol. 13, n.º 16, pág. 9120, 2021.
- [3] M. Barilari, R. J. Siolotto, M. I. Tort y C. Estelrich, «La continuidad del joven rural en su medio y su relación con la mano de obra», *Revista de tecnología agropecuaria*, págs. 54-57, 2018.
- [4] M. A. Díaz, Y. R. Aranza, A. O. Gallo, J. Y. Vega, R. B. Robles et al., «Evolución socioeconómica de la región Caribe colombiana entre 1997 y 2017», Banco de la República-Economía Regional, inf. téc., 2017.
- [5] C. T. Figueroa Jaimes, «La automatización en la agricultura Colombiana¿ Un nuevo reto para el país?», 2022.
- [6] D. DANE, *3er Censo Nacional Agropecuario*, 2016.
- [7] N. Cepal, «Agenda 2030 y los Objetivos de Desarrollo Sostenible: una oportunidad para América Latina y el Caribe», 2018.
- [8] B. O. Nieto y J. C. Rangel, «Sistema de visión artificial para gestión de calidad del Banano Cavendish en etapa de postcosecha», *Revista de Iniciación Científica*, vol. 8, n.º 2, págs. 32-42, 2022.
- [9] A. M. Caicedo Duque, «Sistema de detección y conteo de manos en un racimo de bananos mediante visión artificial», 2021.
- [10] A. G. Durand Tarrillo y F. Abanto Robles, «Comparación de técnicas de visión artificial para detección del defecto del fruto del café», 2021.

-
- [11] F. A. Juanazo Paucar, «Desarrollo de una herramienta informática basada en visión artificial para el reconocimiento del banano para la exportación aplicando procesamiento de imágenes digitales y su simulación en Matlab mediante redes neuronales.», B.S. thesis, Universidad de Guayaquil Facultad de Ciencias Matemáticas y Físicas Carrera ..., 2019.
- [12] E. A. S. Malpartida, *Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot*. Pontificia Universidad Católica del Perú (Peru), 2011.
- [13] A. E. Gamonal Chanco, «Diseño de un sistema por visión artificial para determinar la calidad de mandarinas», 2020.
- [14] R. León, E. Alvarado, K. Arevalo, A. Maldonado y A. Polonio, «Detección y extracción de muestras falladas usando visión artificial y un brazo robótico», *Revista Ciencia y Tecnología*, vol. 16, n.º 1, págs. 21-32, 2020.
- [15] J. A. Cruz-Morales, J. A. Morales-Viscaya, A. I. Barranco-Gutiérrez, A. L. Herrera-May, A. A. Alonso-Ramírez, R. M. Woo-García et al., «Sistema de visión artificial para evaluar cereza de café», *Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI*, vol. 11, n.º Especial2, págs. 205-210, 2023.
- [16] A. de Nova Guerrero, J. J. Guerrero Campo y S. B. Berenguel Baeta, «Detección y segmentación de objetos en imágenes panorámicas»,
- [17] A. S. Llerena Buenaño y M. J. Salazar Villamar, «Automatización de un sistema identificador y posicionador de objetos a través de un Brazo robótico mediante Visión Artificial con Lenguaje Python», B.S. thesis, 2022.
- [18] J. S. Piratova Silva et al., «Control y operación de un brazo robótico virtual mediante Robot Operating System (ROS)», 2021.
- [19] R. Suárez, J. Rosella, M. Vinagreb et al., «Robot Operating System (ROS)», *Este trabajo se ha llevado a cabo por iniciativa y en el seno del Grupo de Trabajo de Innovación de la Asociación Española de Robótica y Automatización (AER).. Accessed: Nov*, vol. 28, 2022.