Auditaje Automático de Góndolas en Supermercados Usando Técnicas de Aprendizaje Profundo y Visión por Computador.

Ing. Emmanuel Morán

Tutor(es):
Prof. Dr. Boris X. Vintimilla Burgos,
Prof. Dr. Miguel A. Realpe Robalino.

Una tesis submitida para el grado de:
Magister en Ciencias de la Computación

17-01-2024

Dedicatoria

Queridos madre, padre, esposa y toda mi familia, con profundo agradecimiento, dedico esta tesis de maestría a cada uno de ustedes. Su amor, aliento y apoyo han sido la fuerza impulsora detrás de cada línea de código y cada logro académico. Gracias por ser mis pilares, mi inspiración y mi mayor fuente de alegría. Este logro es tan suyo como mío.

Declaración Expresa

"Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución: Emmanuel Fernando Morán Barreiro doy mi consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

Emmanuel F. Morán Barreiro

Comité Evaluador

Firmado electrónicamente por:
BORIS XAVIER
VINTIMILLA BURGOS

Boris X. Vintimilla Burgos
Profesor Tutor

Firmado electrónicamente por:
DENNYS FABIAN
PAILLACHO CHILUIZA

Dennys F. Paillacho Chiluiza
Profesor Revisor

Reconocimiento

Este trabajo ha sido apoyado parcialmente por el proyecto ESPOL-CIDIS-11-2022 y Tiendas Industriales Asociadas Sociedad Anónima (TIA S.A.). Una especial mención a TIA, un supermercado minorista de comestibles líder en Ecuador, por brindar acceso a un entorno increíble para la investigación y experimentación durante este proyecto.

Resumen

Los Supermercados son un sector industrial donde se realizan muchas actividades manuales diariamente. Entre las actividades que se realizan existe una que consume mucho tiempo diariamente, y se denomina auditaje de góndolas; ésta engloba varias sub-actividades como: revisión de etiquetas de precios, revisión de estado de productos, perchado de productos, entre otros.

Este proyecto se enfoca en evidenciar la necesidad de una sistema que pueda reportar a los operadores (personal de apoyo) sobre el estado de las góndolas de la tienda, con la finalidad de disminuir el tiempo que toma la revisión de las mismas. Para esto, se expondrá un conjunto de datos formulado para el correcto funcionamiento del proceso, un sistema de adquisición de datos y una flujo de trabajo para resolver el caso de auditaje de góndolas. Finalmente se mostrará resultados sobre una parte del flujo mencionado para evidenciar la robustes del flujo, mostrando posibles mejoras en los bloques que se puedan extender.

Abstract

Supermarkets are an industrial sector where many manual activities are carried out daily. Among those activities there is one that consumes a lot of time daily, and is called shelf auditing; this encompasses several sub-activities such as price tag review, product status review, product display, and others.

This project focuses on evidencing the need for a system that can report to operators (support staff) about the status of the store's shelves, intending to reduce the time it takes to review them. For this, a dataset formulated for the correct functioning of the process, a data acquisition system, and a workflow to resolve the shelves auditing case will be presented. Finally, results will be shown on a part of the mentioned flow to demonstrate the robustness of the flow, showing possible improvements in the blocks that can be extended.

Índice general

Índice de figuras

Índice de tablas

Lista de abreviaciones

GT Verdad Fundamental o Base por sus siglas en inglés (Ground-Truth).

IA Inteligencia Artificial.

IoU Intercepción sobre unión por sus siglas en inglés (Intercepción over Union)..

OCR Reconocimiento Óptico de Caractéres por sus siglas en inglés (Optical Character Recognition)..

QUADS Refiere al etiquetado de objetos con una "caja quadrilateral", donde se considera ángulo..

RBOX Refiere al etiqueta de objetos con una "caja rectangular"..

SOTA State-of-the-art o State of the Art sin guiones. Se refiere, en el ambito cientifico, al Estado de Arte de una tarea..

.

1

Introducción

Hoy en día, el campo de la Inteligencia Artificial (IA) está siendo altamente investigado y desplegado para resolver diferentes problemas para los sectores industriales. La IA ha alcanzado un nivel de madurez alto. Muchos equipos de investigación se esfuerzan a diario para impulsar el estado del arte (SOTA) a alcanzar los límites de precisión teóricos, aún sobrepasando los del ser humano. Por esta razón, se está iniciando una nueva etapa donde la industrialización de la IA es posible de realizar, y en que los modelos desarrollados pueden resolver problemas reales.

Entre los muchos sectores industriales, el de venta minorista, representado grandemente por los supermercados, es uno de los sectores que puede beneficiarse ampliamente de la implementación de IA. Los operadores (personal de trabajo en un supermercado), serían los principales beneficiados de la implementación de la IA, pues al reducir el tiempo que les toma realizar tareas diarias manuales y repetitivas, podrán aumentar el tiempo en tareas más complejas y humanas como la atención personalizada a los clientes, análisis de mejoras y resolución de problemas. Se debe destacar que, agregar sistemas con IA, no implica eliminar plazas de trabajo, sino aumentar la rapidez y eficiencia en el desarollo de las tareas encomendados

Los operadores de los supermercados deben realizar varias tareas manuales diarias, como ya se comentó anteriormente; varias de estas tareas se agrupan en una macro-tarea denominada: Auditaje de Góndolas. Este se puede definir como el proceso de comparar el estado actual de las góndolas con el estado esperado según indica el planograma (modelo visual para la distribución de productos de Supermercado en las respectivas góndolas). Entre las micro-tareas involucradas en el Auditaje de góndolas están: Verificación de precios actualizados, Validación de etiqueta-producto, Validación de estado del producto, Validación de Huecos, entre otros. Todas estas tareas son manuales, mayormente visuales y realizadas por humanos, lo cual implica que puede haber error.

En este trabajo, se realizó una investigación de las necesidades de los Supermercados para el Auditaje de Góndolas. Indicando las micro-tareas más notables y que pueden ser realizadas por un sistema con IA. La implementación de este sistema reducirá el tiempo que el operador utiliza en estas tareas, para

que este recurso pueda ser utilizado en tareas más complejas.

1.1 Antecedentes

Una cadena nacional de supermercados distribuida por todo el país de Ecuador realiza de forma manual su auditaje de góndolas diariamente en las mañanas para validar el estado de sus góndolas. Esto le permite a la cadena poder brindar diariamente a sus clientes un ambiente organizado para realizar sus compras.

Lamentablemente, las tiendas han detectado que la tarea de auditar las góndolas diariamente conlleva un gran consumo de recursos, especialmente en la mañana, antes de abrir al público, requiriendo de 4 operadores por un tiempo de hasta dos horas para revisar y solucionar el estado de sus góndolas solo en las secciones más importantes y/o de alta rotación.

La cadena de supermercados indicó que entre las tareas más prioritarias está la de validar el precio mostrado a los clientes. Esto pues, puede conllevar a un malestar al cliente o pérdidas de ventas.

1.2 Justificación

En el mercado de software actual, no existen soluciones para auditar de forma automática las góndolas que sean completas y guiadas a la tarea prioritaria indicada por el supermercado (esto es, Identificar etiquetas con precios desactualizados). La mayoría de las soluciones se esfuerza en auditar los productos, más que las góndolas propiamente. Otras soluciones son implementadas teniendo en mente un cambio en el trabajo operativo, como lo es usar un operador junto a una aplicación instalada en un teléfono inteligente. Se denota que existe un desalineamiento de las necesidades más importantes del sector que usaría la herramienta. En este trabajo se presentará una solución alcanzable en el tiempo y fuertemente generalizable e integrable a múltiples y variados supermercados.

1.3 Objetivos

- Identificar actividades dentro del auditaje de góndolas que sean manuales y que puedan ser realizadas por un sistema con inteligencia artificial
- Proponer un flujo de trabajo general para generar reportes de errores encontrados durante un proceso automático de Auditaje de Góndolas.
- Diseñar un sistema de adquisición de datos para obtener un conjunto de datos que pueda ser utilizado por el flujo de trabajo propuesto.
- Implementar una parte de la propuesta para identificar de forma automática las etiquetas de precios desactualizadas.

2

Marco Teórico

En este capítulo, se describe el sector industrial de la venta minorista, centrándonos específicamente en los supermercados. Se analizarán algunos de los problemas encontrados durante una investigación profunda en un supermercado, y se identificarán las actividades manuales que se llevan a cabo durante el proceso de auditoría de las góndolas, con el objetivo de evaluar la posibilidad de automatizarlas.

2.1 Los supermercados

A diferencia de las tiendas que venden productos específicos como ropa o electrodomésticos, un supermercado siempre tendrá múltiples categorías para presentar a los clientes. La figura 2.1 muestra algunas de las categorías comunes que se pueden encontrar en un supermercado minorista. El principal objetivo de estos establecimientos es animar a los clientes a realizar en ellos sus compras de hogar completas. Para esto, la tienda debe prepararse y evitar múltiples problemas que pueden desanimar a los clientes a la hora de comprar sus productos. Los escenarios típicos incluyen: no mostrar precios, mal estado del producto, mala presentación, falta de stock, entre otros.

2.1.1 Partes del supermercado

La estructura que tiene un supermercado para brindar un buen ambiente de compra tiene los siguientes elementos principales:

Tienda: La tienda o establecimiento de un supermercado es el espacio físico donde se ubica para ofrecer todos los días sus productos. La figura 2.2 muestra las áreas habituales que se enumeran en el Plano de una tienda, por ejemplo:

- Área de Entrada.
- Área de Pago.
- Área de Atención al Cliente.

(a) Frutas y Vegetales (b) Carnes (c) Electrónicos

(d) Lácteos y Cereales (e) Perfumería (f) Snacks

Figura 2.1: Algunas de las categorías comunes disponibles en un supermercado.

- Área de Ventas (espacio con góndolas).
- Área de Personal (espacio común para los operadores como: comedores, baños de personal, etc.).
- Área de Almacenamiento o Bodega (esto se puede separar en una *zona seca* o mercancía que puede estar a temperaturas normales, como juguetes y cereales, y una *zona fría* o mercancía que necesita estar a una temperatura específica, como carnes, productos lácteos y frutas).

Es común que el departamento administrativo de un supermercado tenga su propio equipo dedicado a evaluar y diseñar los *layouts*, el cual hace referencia a un diseño global para la ubicación de las categorías de productos en las áreas de venta de acuerdo con las estrategias de marketing del supermercado. Este tema no se explorará en profundidad ya que está más allá del alcance de este trabajo.

Góndolas: o también llamadas Estanterias, son estructuras de hierro comúnmente colocadas a lo largo del área de venta de una tienda. Se utilizan para colocar y exhibir los productos en las tiendas de autoservicio como los supermercados. La figura 2.3 muestra una góndola estándar típica que se usa para mostrar los productos colocados y sus partes, así mismo se pueden observar las medidas estándar de alto, ancho y profundidad. Las góndolas tienen varios niveles definidos por sus *bandejas*, las cuales son parte de la góndola que se las puede mover a libertad, y donde se colocan los productos para ser presentados. Las bandejas inferiores son más profundas que las demás bandejas. En el borde delantero de las bandejas se encuentran los *porta etiquetas de precio* de los productos.

Es común que el mismo departamento que se encarga de los layouts, también se encargue de lo que se denomina *planograma*, que es el esquema de cómo se distribuyen los productos en una góndola, esto se profundizará más adelante.

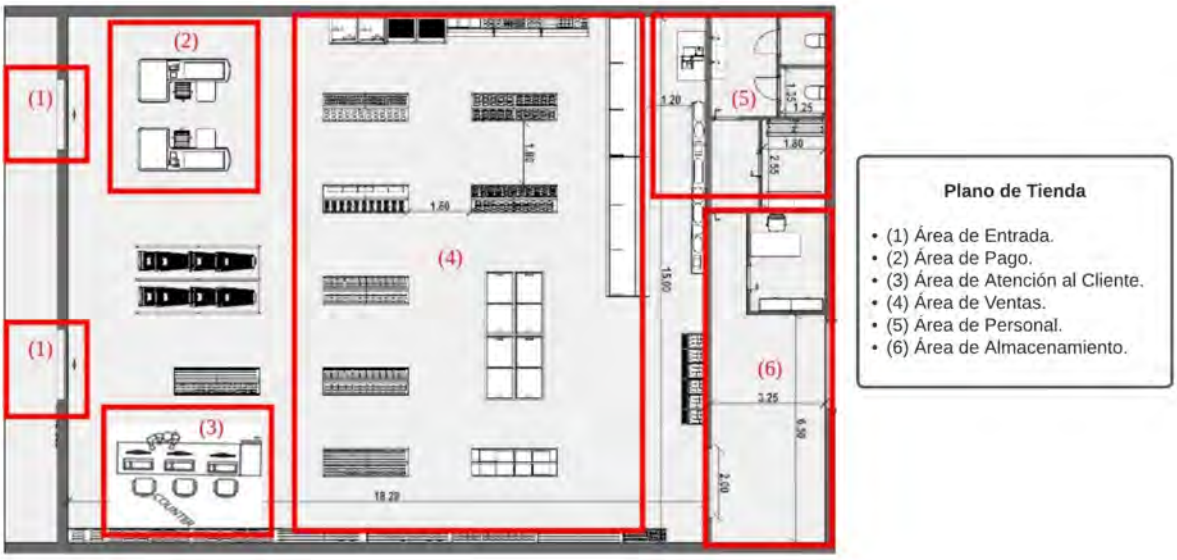

Figura 2.2: Plano de un supermercado con sus áreas.

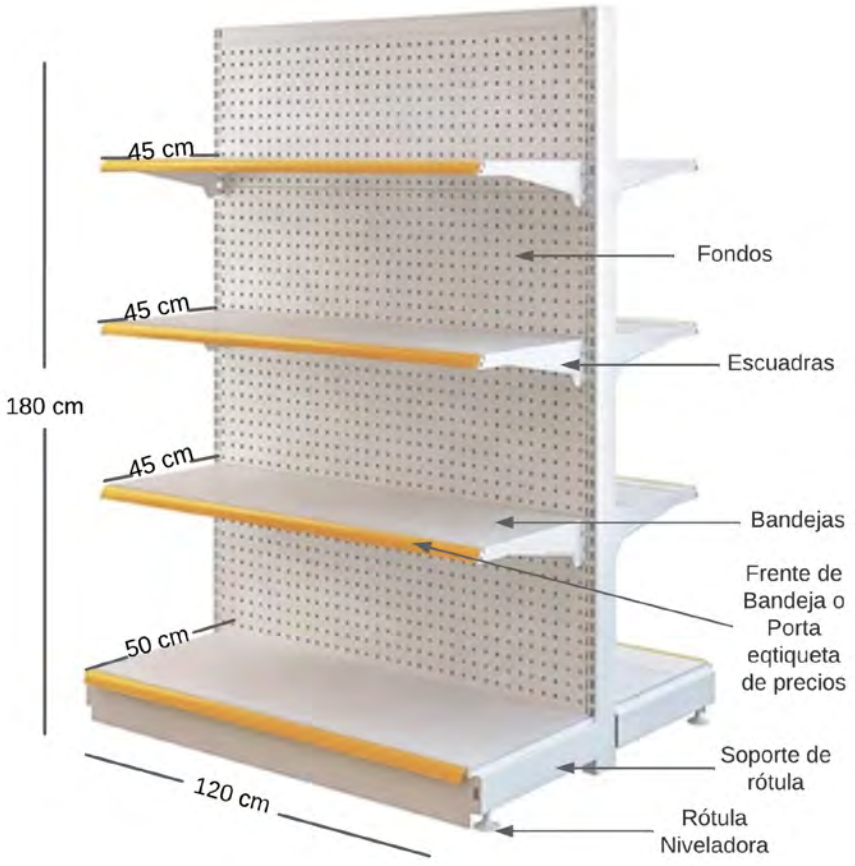

Figura 2.3: Partes y medidas de una góndola estandar de un supermercado.

Figura 2.4: La Etiqueta de precio y su posicionamiento en la bandeja de una góndola según protocolos. (a) Ejemplo de etiqueta de precio con sus partes importantes como: código del producto, precio, descripción y código de barras (b) Posicionamiento de la etiqueta de precio en la parte izquierda inferior del frente del producto en la góndola.

Etiqueta de precio: denominación que se le da a la etiqueta que se coloca cerca del producto en el porta etiquetas para indicar fácilmente al cliente cuál es el nombre y el precio del producto que vé en la góngola. Las etiquetas tienen múltiples dimensiones, colores y diseños, pero tienen información importante comúnmente relacionada con el producto, esto es: código del producto, precio del producto, descripción del producto. La figura 2.4a muestra un ejemplo de una etiqueta (diseño utilizado durante este trabajo). Se puede notar que en muchos diseños de etiquetas se agregan códigos de barra que pueden ser leídos por los operadores usando dispositivos dedicados, para obtener información sobre un producto. Esto suele ser muy útil para verificar los productos, pero hacerlo manualmente conlleva mucho tiempo.

Un dato importante sobre las etiquetas de precio es que deben colocarse en los porta etiquetas de tal forma que queden debajo del *frente* que ocupan los productos. Varios supermercados tienen un protocolo para colocarlos en el lado izquierdo o en el inicio del frente de los productos como se puede ver en la figura 2.4b.

Surtido: Los productos que están posicionados en el área de ventas se clasifican como surtido activo, mientras que el surtido inactivo se refiere a productos que por alguna razón no están presentados en las góndolas. Algunas de las razones más comunes para tener surtido inactivo son: olvido por parte de los operadores, el producto se almacenó intencionalmente para dejar espacio para otro, productos descontinuados, entre otras.

Planogramas: La figura 2.5 muestra un ejemplo de un planograma. El planograma ayuda a los operadores de la tienda a posicionar el surtido en los góndolas. Es probable que las tiendas pequeñas no tengan reglas complejas para colocar los productos, sin embargo, en los supermercados más grandes, tener un planograma ayuda a tener un orden y comprensión de lo que sucede con los productos si suben o bajan de las bandejas. Existe toda una teoría sobre el posicionamiento de los productos en las góndolas [1, 2, 3, 4]. En las tiendas existe personal encargado específicamente de validar que estos planogramas se estén utilizando correctamente. Es importante seguir el planograma porque puede

Figura 2.5: Ejemplo de planograma de una góndola.

haber contratos con proveedores, que han pagado extra para que sus productos estén en una mejor posición o tengan un frente de bandeja más ancho; otra razón para seguir los lineamientos del planograma es la experimentación de que productos se venden mejor estando cerca o lejos de otro, existen teorías que esto puede aumentar o disminuir su venta [5, 6], y esto solo se puede validar si se posicionan de acuerdo con el planograma enviado a la tienda.

2.2 Problemas de los supermercados

Muchos problemas en los supermercados minoristas han sido identificados y reportados en la literatura [7] [8] [9]. Este trabajo se centrará en los problemas relacionados con la presentación de los productos en las góndolas. Esto implica el posicionamiento de los productos y etiquetas de precios que es realizado por los operadores y que inevitablemente tendrá errores.

2.2.1 Etiquetas de precios obsoletas

Este problema se refiere a etiquetas de precios (impresas o digitales) que están desactualizadas (precio mostrado no es el que será cobrado en la caja de pago) o en mal estado (rotas, manchadas, mal impresas, etc.). Las etiquetas de precio son uno de los objetos más importantes en las góndolas, ya que permiten saber qué producto se está exhibiendo, su descripción, peso o tamaño y lo más importante, el precio. Si las etiquetas de precios están desactualizadas, es imposible que los clientes no identifiquen correctamente el precio real del producto. Un precio desactualizado en la etiqueta de precio puede

tener dos repercusiones negativas:

- Si el precio real es más bajo que el mostrado (por ejemplo, un producto que debería tener una oferta especial), probablemente no sea atractivo para los clientes y, por lo tanto, las ventas esperadas por la disminución en el precio no se logren alcanzar, perdiendo ventas y margen de ganancia.
- Si el precio real es más alto que el mostrado en la etiqueta de precio, el cliente puede sentirse engañado al momento de pagar en la caja. Esto último podría tener repercusión legal ya que puede considerarse una estafa al cliente.

Una solución convencional podría ser etiquetas de precios digitales/electrónicas [10]. Estas son etiquetas de precios que pueden cambiar instantaneamente su contenido (precio, descripción, código, etc.) y puede ser realizado para múltiples etiquetas en un solo proceso. Lamentablemente implementar este tipo de soluciones es muy costoso y la mayoría de los minoristas no pueden afrontarlo. También se podría pensar en usar la tecnología RFID para obtener fácilmente información de una etiqueta de precio, pero esta tecnología no es convencional para este uso en particular, ya que normalmente se usa para identificar productos en la cadena de suministro [11, 12] y facilitar el inventario en categorías con alto margen de ganancia como ropa y botellas con bebidas alcohólicas como whisky [13, 14].

2.2.2 Surtido inactivo

Surtido Inactivo es la denominación que se le da a los productos que deberían exhibirse en el área de ventas (espacio donde están las góndolas y se exhiben los productos), pero por alguna razón no están presentados. Algunas de las razones para que esto suceda son: existencias agotadas, existencias aún no se encuentran físicamente en la tienda, decisión de la tienda de guardar el producto en la bodega. El surtido inactivo tiene el potencial de ahuyentar a los clientes, los cuales al no encontrar todos los productos que desean, es probable que en un futuro opten por no ingresar a la tienda por el pensamiento de que no está bien surtida.

2.2.3 Surtido faltante

Denominación para un producto parcialmente exhibido en las góndolas. Esto se considera pésimo para la presentación porque la góndola aparenta estar vacía. Esto es diferente del surtido inactivo porque el producto se muestra en la bandeja de la góndola (área del estante para el producto) pero está parcialmente vacío. Esto puede ocurrir por muchas razones, como falta de existencias ó reposición aún no completada. Este último es el error más común y se puede corregir revisando manualmente todos los productos en las góndolas y anotando los faltantes para luego proceder a llenarlos, luego de sacar los productos de la bodega.

2.2.4 Surtido en mal estado

El estado de un producto es cómo se lo muestra en la góndola; también puede considerar como el estado del empaque del producto que puede estar roto, rasgado, lacerado u otros que indiquen que su estado ha sido malogrado. Si un producto se encuentra en mal estado podría llevar a un cliente a dimitir de hacer la compra e incluso a no volver a la tienda por el pensamiento de que los productos son

maltratados. Esto podría ser culpa de los operadores que no han seguido los protocolos de la tienda o quizás de algunos niños jugando en la tienda.

2.3 Auditaje de góndolas

La auditoría de góndolas se puede definir como el proceso de comparar el estado actual de las góndolas con el estado que deberían tener según el planograma del supermercado. Esta comparación es necesaria por muchos aspectos, entre ellos, que el área de ventas esté acorde a la planeación para poder validar el cumplimiento de contratos con proveedores que han pagado extra por posicionamiento estratégico de sus productos en las bandejas, o para poder realizar un análisis de mercado, por ejemplo, cómo se comportan ciertos productos si se colocan más cerca unos de otros o si se alejan.

Esta comparación es realizada tradicionalmente por los operadores de supermercados, y como todo proceso repetitivo realizado por un ser humano, es propenso a errores y requiere mucho tiempo. Las comparaciones tienen un carácter visual, es decir, los operadores necesitan utilizar su visión y entender el concepto de góndola ordenada para poder identificar errores, anotarlos y luego solucionarlos.

Durante la auditoría de góndolas, se deben validar los siguientes criterios:

- Presencia de todos los productos en cada bandeja de la góndola según el planograma.
- Presencia de las etiquetas de precio de los productos en la posición correcta.
- Concordancia entre producto y su etiqueta de precio.
- Precio del producto actualizado en la etiqueta de precio.
- Presencia de espacios en las góndolas.
- Estado del producto. [1]

Para realizar estas validaciones, los operadores utilizan su capacidad innata de vísión. Mientras que para automatizar este proceso y permitir que pueda ser realizado por una computadora, se requieren dos tareas fundamentales: Detección de Objetos y Reconocimiento de Objetos. Estas dos tareas son muy amplias y aún requieren investigación para poder avanzar en el estado del arte. En el presente trabajo, se explica el estado del arte de la tarea de detección de objetos por ser la principal tarea para el proceso.

2.4 Detección de objetos

La detección de objetos es la tarea de localizar e identificar objetos presentes en una imagen o video [15]. Es una de las tareas con mayor tendencia dentro del campo de la inteligencia artificial y la visión por computadora. Actualmente se utiliza en algunos proyectos como conteo de personas, detección de peatones, conteo de frutas en plantaciones y otros [16, 17, 18, 19, 20]. La detección de objetos localiza e identifica los objetos en la escena que se le presente. Tiene la caracteristica de que la escena es independiente pues se entrena para múltiples y variadas escena con lo cual logra generalizar de

[1] Esto podría corresponder a: cantidad correcta de frentes, productos organizados u otros que sean parte del protocolo del supermercado.

manera efectiva la localización e identificación de los objetos, un ejemplo puede ser que estos algorítmos pueden detectar personas en diferentes ángulos, partes de la escena, y con escenas variadas. Normalmente las escenas con las que son entrenadas estos modelos no superan las decenas de objetos por imágen, pero existen aplicaciones donde las escenas tienen una cantidad de objetos que superan las centenas. A estos escenarios se los denomina: Altamente Densos. Estos escenarios aumentan la dificultad pues en muchos casos los objetos son muy similares entre sí.

En esta sección se explicarán las métricas más comunes para la detección de objetos, los conjuntos de datos más conocidos para esta tarea, y luego se enfocará en una explicación sobre la tarea de detección de objetos en ambientes altamente densos, dando a conocer algunos conjuntos de datos, y el estado de arte actual.

2.4.1 Métricas para detección de objetos

La tarea principal de la detección de objetos es de localizar un objeto en una escena, para lograr esto, se define un recuadro (RBOX) rectangular, que será de ayuda para localizar el objeto en la escena. Del conjunto de datos se tiene RBOXs que fueron etiquetados manualmente, dando una gran certeza de ser completamente reales en la escena; a estos RBOXs etiquetados manualmente se los conoce como la verdad fundamental (Ground Truth o GT, por sus siglas en inglés). Dado esto, la tarea se puede simplificar a predecir los recuadros más similares a los originales durante el entrenamiento, y por otro lado evitar predecir recuadros que no son los originales. Esto se puede medir con las métricas de Precisión (P) y la Recuperación (R). Estas son las métricas que se utilizan para medir el desempeño de un algorítmo para la tarea de detección de objetos [21]. Para poderlas definir correctamente, se debe revisar los conceptos que son compartidos entre ellas:

- Verdadero positivo (TP): Una detección presente en el GT y detectada correctamente.
- Falso positivo (FP): Una detección incorrecta de un inexistente objeto o una detección fuera de lugar de un objeto existente
- Falso negativo (FN): Una detección presente en el GT pero no detectada.

Se debe señalar que en este contexto lo reconocidos como Verdadero Negativo (TN) no es aplicado por la infinidad de detecciones que no deben detectarse dentro de la escena.

Para poder categorizar a una detección como correcta o incorrecta se utiliza la operación de *Intersección sobre Unión* (ó IoU, por sus siglas en inglés). Esta es la medición basada en un coeficiente de similitud denominado índice de jaccard, el cual se usa para medir el área de superposición entre las detecciones predecidas y las detecciones del GT o Reales. La formula de este algorítmo se lo puede visualizar en la figura 2.6. Comparando el IoU de las detección con ayuda de un umbral definido *'T'*, podemos clasificar una detección como correcta o incorrecta. Si la Intersección sobre la unión entre un RBOX perteneciente al GT y un RBOX detectado es mayor o igual al umbral definido ($IoU_{GT-DET} \geq T$), entonces la detección es correcta, caso contrario la detección será incorrecta. Con esto podremos generar nuestra matriz de confusion y obtener nuestros resultados de Precision y Recuperación de la siguiente manera:

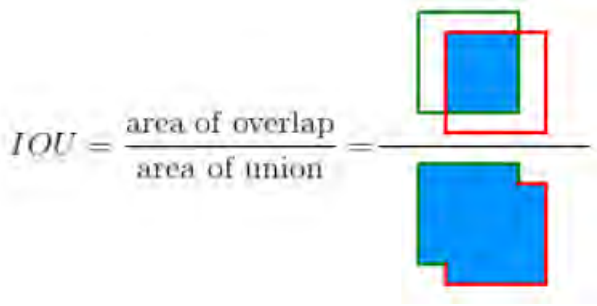

Figura 2.6: Apreciación visual de la operación de intersección sobre unión.

Tabla 2.1: Conjuntos de datos comunes para detección de objetos en ambientes poco y altamente densos.

Propiedad \ Conjunto de datos	PASCAL-VOC	COCO	CARPK	SKU110K
Número de Imágenes	22,531	328,000	1,448	11,762
Número de Objetos	61,059	2,525,600	89,777	1,733,718
Densidad	2.71	7.7	62	147.4

$$P = TP/(TP + FP) \tag{2.1}$$

$$R = TP/(TP + FN) \tag{2.2}$$

Con lo antes mencionado, se pueden sacar las siguiente métricas que serán de ayuda tanto para ambientes comunes y altamente densos:

- **Precisión Promedio (PP)**: Esta métrica se obtiene sacando el promedio de las precisiones de los ejemplos. Se acostumbra a utilizarla junto a un valor de IoU para saber que tan estricta es la métrica de evaluación de las detecciones. Ejemplo: PP[0.5] significa Precisión Promedio con IoU mayor o igual a 0.50. Normalmente en la literatura se encuentra como AP[0.50] por sus siglas en inglés (Average Precision). También es común ver la métrica mAP[0.50] la cual realiza el promedio entre las clases.
- **Recuperación Promedio (PP)**: Esta métrica se obtiene sacando el promedio de las recuperaciones de los ejemplos. Se acostumbra a utilizarla junto a un valor de IoU para saber que tan estricta es la métrica de evaluación de las detecciones. Ejemplo: RP[0.75] significa Precisión Promedio con IoU mayor o igual a 0.75. Normalmente en la literatura se encuentra como AR[0.75] por sus siglas en inglés (Average Recall). También es común ver la métrica mAR[0.5] la cual realiza el promedio entre las clases.

Finalmente, también es habitual ver estas dos métricas promediadas para múltiples valores de IoU, normalmente [0.50, 0.95], que significa que se realiza el promedio del valor final de la métrica para cada valor de IoU, esto se lo conoce como métrica estricta.

2.4.2 Conjuntos de datos para detección de objetos

Una parte muy importante de los algorítmos de aprendizaje automático son los datos. Estos se alimentan directamente de los datos para aprender patrones y generalizar reglas con el fin de realizar las tareas para las que fueron capacitados.

Al día de hoy existen varios conjuntos de datos para la tarea de detección de objetos. Muchos son generados de manera personalizada para objetos poco comunes. En varias ocasiones se ha utilizado para herramientas en la industria [22, 23], con lo cual se generan sistemas que aportan a la selección de caminos en las bandas de manofactura para eliminar aquellas herramientas con defectos.

Los conjuntos de datos para esta tarea suelen estar divididos por un lado en la imagen (contiene la escena donde se detectarán los objetos) y las etiquetas (también llamadas labels en inglés) y que normalmente tienen la nomenclatura de: [id-objecto, X1, Y1, X2, Y2]. Donde id-objeto es el índice del tipo de objeto; (X1, Y1) y (X2, Y2) se refieren a las esquinas superior izquierda e inferior derecha de la detección, respectivamente.

Entre los conjuntos de datos públicos para la detección de objetos, con objetos más comunes, y también más utilizados, están COCO [24] y PASCAL-VOC [25]. Estos son ampliamente usados para la detección de objetos y son comúnmente usados como un evaluador del modelo. En el cuadro 2.1 se pueden observar algunas características de estos conjuntos de datos. Para un ambiente simple, sin muchos objetos en la escena, estos conjuntos de datos son perfectos para entrenar el modelo y realizar múltiples experimentos como transferencia de aprendizaje. Pero, para escenarios más completos, estos conjuntos de datos, no le entregan a los modelos de aprendizaje de máquina la suficiente experience para reconocer los objetos en las escenas. Estas escenas son conocidas como: Ambientes altamente densos.

2.4.3 Ambientes altamente densos

Se define como ambientes altamente denso, a aquellos escenarios donde los objetos están muy cercanos unos de otros, la concentración de los objetos es muy alta con respecto a la imágen, y también los objetos son muy semejantes. Algunos de los ambientes considerados altamente densos pueden ser: parqueaderos para el objeto carro, naturaleza para objeto planta, y supermercados para objeto producto. En la sección 2.4.2 se habló de conjuntos de datos para detección de objetos. Los mencionados COCO y PASCAL-VOC lamentablemente no tienen la densidad por imagen necesaria para ser considerados ambientes altamente densos. En el cuadro 2.1 se puede observar conjuntos como CARPK [26] y SKU110k[27]. Estos conjuntos tienen imágenes de escenas con varios objetos. Es importante mencionar que los entornos altamente densos no son extraordinarios ni están diseñados en laboratorio para pruebas exhaustivas. De hecho, son entornos bastante comunes en la vida real o en la naturaleza. Los entornos altamente densos son, sin duda, el reto que muchos algoritmos deben pasar en algún momento (al menos para el tipo de dato imágenes) para poder llamarse robustos. En estos ambientes es común encontrar problemas con objetos parcialmente ocluidos y objetos muy cercanos, los cuales son muy problemáticos para la detección de objetos [28].

Para este trabajo haremos un enfoque directo al último ambiente mencionado, el cual es los supermercados. Este ambiente es considerado altamente denso pues los objetos están muy cerca uno de otro y son muy semejantes unos a otros. En una misma imagen pueden haber centenas de objetos. En la figura 2.7 se puede observar ejemplos del conjunto de datos SKU110k.

Figura 2.7: Ejemplos de imágenes del conjunto de datos SKU110K, en ambientes altamente densos de supermercados.

2.4.4 Detección de objetos en ambientes altamente densos

En 2019, Goldman et.al. [27] propuso el primer conjunto de datos para entornos altamente densos para el sector minorista, denominado SKU110K[27]. En el cuadro 2.1 también se puede ver que la densidad de objetos/imagen de este conjunto de datos es más del doble que la de CARPK.

Hoy en día, existe un conjunto de datos público reciente llamado UniDet [29]. Este conjunto de datos es una copia de SKU110K con la distinción de aumentar en 500 imágenes capturadas con diferentes sensores y en diferentes ángulos (utilizado para el conjunto de datos de prueba y para probar las condiciones de dominios cruzados). También cambia el estilo de las anotaciones a un modelo llamado QUADs, que son anotaciones cuadriláteras que se acoplan mejor a los objetos que las típicas anotaciones tipo RBOX (caja rectangular).

Una particularidad de las metodologías aplicadas con el conjunto de datos SKU110K es que las detecciones son independientes de la clase. En otras palabras, buscan únicamente localizar los objetos, sin la sobrecarga de identificarlos o reconocer a qué clase pertenecen. Solo la tarea de detectar correctamente todos los objetos en una escena es compleja, y en entornos densos este objetivo se vuelve mucho más complicado. En estas escenas podrían surgir algunos problemas como luminosidad, oclusión, etc. La figura 2.7 muestra ejemplos de escenarios para el conjunto de datos SKU110K, donde cabe señalar que los ángulos de captura se suman a la lista de problemas de este conjunto de datos pues existe una inclinación del sensor de captura con respecto al perpendicular de las góndolas.

Para ambientes altamente densos existen varios modelos que escogen un ambiente y lo atacan para resolverlo de la mejor manera posible. En este trabajo hablaremos más a profundidad del ambiente altamente denso de los supermercados enfocandonos en el uso del conjunto de datos SKU110k por ser el primer impulsador para trabajar en esta linea. La descripción del estado del arte o SOTA de este trabajo se centrará en los datos obtenidos por trabajos anteriores que presenten resultados usando el conjunto de datos SKU110K, que es el único conjunto de datos público que tiene en cuenta un entorno altamente denso para el comercio minorista y, por lo tanto, de gran interés para la auditoría de góndolas. Como muchas otras evaluaciones de detección de objetos, se definen las siguientes métricas:

- **mAP[0.75]**, Precisión media promedio con IoU=0.75.
- **mAP[0.5:0.95]**, Precisión media promedio. Usa IoU con valores de 0,50 a 0,95 incrementados en

Tabla 2.2: Estado del Arte para Detección de Objetos en los Supermercados usando el conjunto de datos SKU110k. mAP: mean average precision o precision media promedio. AR: average recall o recuperación promedio. *usa el promedio de la métrica evaluada con IoU entre [0.5:0.95] aumentando en 0.05.

Métrica / Modelo	**mAP[0.75]**	**mAP***	**AR300[0.5]**	**AR300***
Faster-RCNN [30]	0.010	0.045	-	0.066
YOLO9000 [31]	0.073	0.094	-	0.111
RetinaNet [32]	0.389	0.455	-	0.530
RetinaNet + GD [33]	0.552	0.512	0.917	0.582
RetinaNet + Soft-IoU + EM-Merger [27]	0.569	0.514	0.872	0.571
RetinaNet + GL [33]	0.562	0.521	**0.931**	0.596
YoloV3 [34]	0.568	0.554	-	0.562
Dynamic Refinement Network [35]	**0.640**	0.569	-	**0.635**
Res2Net-101+HRNet-W32[36]	-	0.584	-	-
Cascade R-CNN + SS [37]	-	0.587	-	-
RetailDet [38]	-	**0.603**	-	-

0,05 cada vez, para evaluar la precisión y luego tome la media de este conjunto.

- **AR300[0.5]**, Recuperación promedio. 300 significa que está limitado a usar las 300 mejores detecciones. Usa *confianza de ser un objeto* como una variable para ordenar de mayor a menor. Utiliza IoU=0.5.

- **AR300[0.5:0.95]**, Recuperación promedio. 300 significa que está limitado a usar las 300 mejores detecciones. Usa *confianza de ser un objeto* como una variable para ordenar de mayor a menor. Utiliza IoU con valores de 0,50 a 0,95 incrementados en 0,05 cada vez, para evaluar la recuperación y luego tomar el promedio de estos valores.

El cuadro 2.2 muestra varios modelos con las métricas indicadas para el conjunto de datos SKU110k. Es fácil identificar que los modelos Faster-RCNN [30] y YOLO9000 [31] tienen valores de precisión muy reducidos. Esto se debe a que estos modelos fueron desarrollados sin tener en consideración a ambientes altamente densos. Sus implementaciones no dan la factibilidad para ser escalables para estos ambientes y se empezó el desarrollo de estratégias para solventar los desafios encontrados en estos ambientes. Para el caso particular del conjunto de datos SKU110k se empezó teniendo pruebas usando el modelo RetinaNet [32], este modelo propuso una novedosa pérdida focal centrando el entrenamiento del conjunto en casos difíciles evitando que la gran cantidad de falsos negativos abrumen el detector, remodelando la pérdida de la entropía cruzada estándar. Se logró conseguir una base de pruebas apropiado logrando una precisión de 0.455 sobre este conjunto de datos. Luego el autor del conjunto de datos publicó sus resultados usando el modelo de RetinaNet con dos cambios: Agregó una capa de Soft-IoU para estimar el índice Jaccard, y una unidad EM-Merger que convierte detecciones y puntuaciones de la capa Soft-IoU en una mezcla de gausianos y resuelve detecciones superpuestas en escenas empaquetadas. El mismo autor luego publicó otros pesos donde aumentó la precisión aumentando desde 0.492 que consigió al inicio hasta 0.514.

Más adelante, [33] propone una mejora a RetinaNet usando una red de decodificador gaussiano (GDN, Gaussian Decoder Network) y una red de capa gaussiana (GLN, Gaussian Layer Network). GDN en lugar de utilizar la red piramidal de funciones como decodificador utiliza un decodificador separado que predice los conjuntos de gaussianos 2D de cada objeto en la imagen, miéntras que GLN propone una capa gaussiana en la arquitectura RetinaNet con menores parámetros y mayor precisión, la arquitectura de aprendizaje multitarea con codificador y decodificador compartido y se aplica una pérdida gaussiana adicional en la salida de la subred gaussiana. Estas mejoras dieron como resultado una precision de 0.512 y 0.521, dandole una pequeña pero considerable mejora al modelo publicado por el autor de SKU110k.

Luego, [35] presenta un modelo denominado Red de refinamiento dinámico, el cual consta de dos componentes novedosos: un módulo de selección de características (FSM, a feature selection module) y un cabezal de refinamiento dinámico (DRH, a dynamic refinement head). El FSM permite a las neuronas ajustar los campos receptivos de acuerdo con las formas y orientaciones de los objetos objetivo, mientras que el DRH permite a nuestro modelo refinar la predicción dinámicamente de manera consciente del objeto. Cabe destacar que este trabajo aumentó el SKU110k creando cuadros delimitadores orientados. Este modelo llevó la precision hasta 0.569, en este trabajo también presentan resultados del modelo YoloV3 sobre los datos de SKU110k teniendo resultados de 0.554 en presición.

Posteriormente, aparece una solución que consta de un ensamble de dos modelos. Ambos basados en la selección adaptativa de muestras de entrenamiento (ATSS, Adaptive Training Sample Selection). El modelo 1 adopta HRNet-W32, HRFPN y ATSS. El modelo 2 emplea Res2Net-101, FPN Balanceado, y ATSS. Para el ensamble utilizan WBF (Weighted Boxes Fusion). Este modelo tuvo una precisión sobre SKU110k de 0.584.

En el 2021 apareció una solución presentada en un reporte técnico, el cúal indica que se adoptó una estrategia de cultivo aleatorio modificada y un modelo Cascade R-CNN optimizado que tuvo una precisión de 0.587 sobre el conjunto de datos SKU110k.

El estado del arte actual es el modelo presentado por [38] con una precisión media del 0.603. Este modelo utiliza un conjunto de datos similar al de SKU110k mejorado mediante 500 imágenes extras para pruebas del modelo. En 4 años, el valor de precisión para la tarea de detección de objetos en ambientes altamente densos de supermercados, ha aumentado aproximadamente un 11%. Esto confirma lo antes mencionado que existe un alto interés en solucionar problemas para éste sector.

3

Propuesta

En este capítulo se definirá una propuesta para solucionar el problema de la auditoria de góndolas en supermercados de forma automática. Se iniciará proponiendo un conjunto de datos junto a un sistema de adquisición, para finalizar exponiendo un flujo de trabajo como solución a varios problemas mencionados anteriormente.

En la figura 2.7 se pueden ver ejemplos de imágenes en el conjunto de datos SKU110k. Pero estos no son suficientes para poder continuar, debido a su estrategia de adquisición, la cual repercute en la resolución del objeto etiqueta de precio. Para seguir avanzando es necesario cubrir más necesidades de los supermercados, y varias de estas se basan en poder obtener información de las etiquetas de precio. En la figura 3.1 se muestra un ejemplo de las etiquetas de precios de los conjuntos de datos SKU110k y UniDet respectivamente. Se puede observar que los precios son los únicos textos de la etiqueta relativamente apreciables, ya que normalmente es el texto más destacado para que el cliente lo pueda notar rápidamente. Dado que el precio es lo único legible, estas imágenes no permitirían reconocer a qué productos representan las etiquetas.

Un alcance actual presentado por [29] tiene una forma particular de validar qué producto está presente en el proceso de detección usando OCR en los textos del empaque del producto para obtener una cadena de textos que luego pasan por un proceso de emparejamiento para identificar la descrip-

Figura 3.1: Ejemplo de etiquetas de precios de los datasets SKU110k (izquierda) y UniDet (derecha).

ción de la base de datos de productos que más se parezca y reconocer el producto. Sin embargo, esto nuevamente no proporciona la certidumbre de emparejamiento con la etiqueta de precio, y particularmente puede dar ciertos fallos en caso de que los empaques del producto se encuentren deteriorados, así mismo en secciones como vidrios y carnes será mucho más complejo implementar este tipo de soluciones ya que el empaque viene con poco o nada de textos para identificar el producto o simplemente no tienen un empaque.

En este trabajo se considera a la etiqueta de precio como uno de los objetos más importantes dentro de las góndolas, pues no solo presentan textos para reconocer el producto y su precio, sino que también pueden dar una idea de organización de los productos en las bandejas de las góndolas. Muchos trabajos se enfocan directamente en detectar y reconocer los productos, pero no profundizan en la necesidad de organización de los productos, la cual puede ser llevada a cabo únicamente con las etiquetas de precio.

Está claro que tener conjuntos de datos como SKU110k generó un gran interés, sin embargo, es hora de volver a evaluar si estos conjuntos de datos pueden proporcionar mejores resultados y ser parte de una solución completa y viable para el sector minorista.

3.1 Conjunto de datos

Los datos de entrada requeridos para la propuesta presentada son los siguientes:

- **Planometrías:** La planometría (datos usados para generar los planogramas) debe ser entregada de forma estructurada tal que se listen los productos y número de frentes que ocupa en cada bandeja de la góndola. Esta información podría recopilarse directamente de una base de datos si la empresa es lo suficientemente madura.
- **Base de Datos Maestra:** Una base de datos para consultar información de los productos como precios, descripciones, códigos de barra, entre otros.
- **Imagen RGB-UHD:** Imágenes similares a la figura 3.2a. Son imágenes RGB (3 canales rojo, verde y azul) pero debe ser capturadas con ayuda de cámaras de alta definición, preferiblemente 4k o superior.
- **Imagen de profundidad:** Imágenes similares a 3.2b. Deben ser capturadas mediante una cámara 3D que permita obtener la profundidad medida desde la lente hasta los objetos.
- **Información de posición:** Información de posición o ubicación relativa al sitio de recolección. Se puede entregar en formato [x,y,z] donde x,y,z son valores decimales que representan un punto en el espacio.

Todos estos datos que se obtienen tienen una premeditada necesidad dentro del flujo de trabajo que se presentará más adelante. Los primeros 2 elementos del conjunto de datos son obtenibles por parte del supermercado. Es decir, que el supermercado debe ser lo suficientemente maduro en sus sistemas para poder implementar este tipo de soluciones.

Por otro lado, los 3 elementos faltantes, son datos únicamente obtenibles con un recorrido por la tienda. La información de posición se refiere a una localización relativa a un sistema de referencia diseñado

(a) (b)

Figura 3.2: Ejemplos de imágenes del conjunto de datos. (a) Imagen del tipo RGB-UHD obtenida con una cámara de alta resolución. (b) Imagen de profundidad obtenida con una cámara 3D.

para la tienda. Las imágenes de tipo RGB-UHD y de profundidad son obtenidas por cámaras especializadas. Es fácil intepretar que estás cámaras requieren estar muy cercanas, pues van a requerir capturar la misma escena o por lo menos lo más similar posible, considerando el campo de visión. En la siguiente subsección se extenderá a detalle como el sistema de adquisición de datos fue diseñado y como recolecta la información.

3.2 Diseño del sistema de adquisición de datos.

Tomando en consideración los 3 elementos no obtenibles de las bases de datos de los supermercados, se presenta ahora un diseño de sistema de adquisión de datos. Este sistema es pensado en ser autónomo y que pueda obtener las imágenes de las góndolas de las tiendas, considerando la altura de las mismas.

En la figura 3.3 se observa la propuesta de diseño para el sistema de adquisición de datos. El sistema refiere a un robot autónomo con 3 tipos de estructuras principales:

- **Estructura de Base:** se denomina a la base del robot que consta de una equipo de procesamiento para la navegación autónoma del robot, bateria, reguladores, motores, llantas y sensores como encoders para odometría en las llantas y lidar.
- **Estructura de Torre:** se denomina a una estructura de aluminio (u otro material más ligero pero

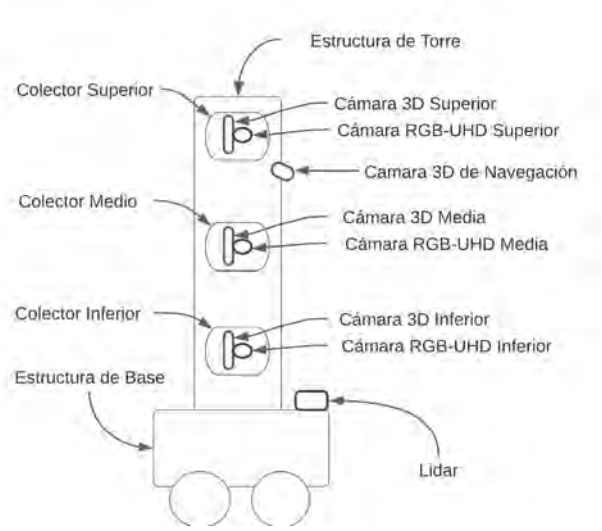

Figura 3.3: Diseño estructural del sistema robótico autónomo, incluyendo sus colectores internos; Superior, medio e inferior.

igual o mayor resistente) que se posa encima de la base con la finalidad de darle alojamiento a los colectores. Esta estructura puede ser agrandada en altura dependiendo de la altura de las góndolas de los supermercados.

- **Estructura de Colector:** se denomina a pequeñas estructuras colocadas dentro de la estructura de torre y que están compuestas de una unidad de procesamiento, una cámara 3D y una cámara RGB de alta resolución. En la figura 3.3 se pueden observar solamente 3 colectores cada uno con sus debidas cámaras, sin embargo, si es necesario pueden colocarse más.

El diseño propuesto tiene como idea principal ser autónomo para evitar que exista más carga laboral a los operadores. Este sistema tiene el potencial de ejecutarse durante las horas muertas de la tienda (horario de la madrugada) y procesar los resultados para que al inicio del horario laboral de la tienda ya existan los reportes necesarios listos para ser utilizados para ejecutar acciones inmediatas.

El proceso de adquisición de los datos por parte del sistema sería autónomo y realizado de manera pausada para poder adquirir los datos lo más nítidos posibles. Es decir, el sistema realizaría pequeños *pasos* frente a las góndolas de la tienda, capturando las imágenes y datos de posicionamiento. Este proceso debería realizarse 2 veces por cada pasillo, dado que el sistema solo posee colectores en un lado la estructura de la torre. Poner cámaras de ambos lados de la estructura de la torre, para evitar doble recorrido por los pasillos, implicaría aumentar la complejidad durante el recorrido y el costo del sistema de adquisición de datos; además, si los pasillos son muy estrechos o muy amplios de igual manera el sistema requeriría dar doble recorrido para recolectar los datos lo suficientemente cerca de las góndolas para que exista nitidez en los datos que se capturan.

Cabe mencionar que el espaciado entre las cámaras de cada colector es lo más reducido posible, y apreciablemente el mismo siempre, dado que las cámaras se posarán sobre la estructura del colector. Esto permite reducir calculos posteriores sobre calibración de los equipos. Así mismo, las estructuras de los colectores están espaciadas verticalmente con la misma distancia y considerando que los campos de visión de las cámaras se interlapen, a estas zonas de interlape se la denominará *redundancia ver-*

tical. Esto último es requerido para evitar pérdida de información. Estas zonas de interlapado generan redundancia entre los objetos que se pueden detectar, como los etiquetas de precio o productos, que se pueden detectar en la zona de interlapado de dos cámaras. Debido al modo de recolección (step-by-step o paso a paso) también existirá lo denominado *redundancia horizontal*. Más adelante se va a explicar estas redundancias de forma más profunda. Es importante indicar que entre el conjunto de objetos redundantes siempre habrá un objeto que tenga la mejor resolución y es necesario obtenerlo del conjunto formado de redundancias para poder continuar el proceso y eliminar la sobrecarga de procesos sobre otros objetos redundantes.

3.3 Propuesta de flujo de trabajo

En la figura 3.4 muestra la canalización de la solución completa propuesta en este trabajo, para el problema de auditoría de góndolas.

Los bloques de color azul son las entradas al sistema y son los elementos considerados en el conjunto de datos presentados anteriormente. En la figura 3.4 se pueden apreciar en la parte superior los 3 elementos que se obtienen con el sistema de adquisición de datos, indicando que son vitales para iniciar el proceso. En la parte inferior se encuentran los dos elementos obtenibles del supermercado, y que son guiados a ser datos que debe actualizar el supermercado constantemente para que sea validado de forma automática por la solución propuesta. Estos elementos se encuentran cerca de las salidas del sistema pues son utilizados mayormente para contrastar la información recolectada y la información que debería estar presente.

Los bloques de color gris representan procesos intermedios e interconectados para llevar a cabo, durante diferente etapas y de manera paralela, el procesamiento de los datos recolectados por el sistema de adquisición. Estos bloques involucran el uso de algoritmos de inteligencia artificial, Visión por computadora, Aprendizaje Profundo, Agrupamiento espacial, etc. Entre las tareas que se mencionarán más adelante se tiene: detección de objetos, reconocimiento de objetos, reconocimiento de texto, entre otros.

Los bloques de color amarillos representan procesos de validación y estimación que utilizan los datos procesados por los bloques de color gris para finalmente manipularlos y filtrarlos para poder crear los reportes finales o salidas del sistema.

Los bloques de color verde son las salidas del sistema y representan reportes guiados a solucionar los problemas mencionados en el capítulo anterior como etiquetas obsoletas, surtido no activo, productos mal perchados, entre otros.

Cabe mencionar que el flujo propuesto no pretende agregar nuevo hardware o procesos a las tiendas. Por tal razón todos los elementos del conjunto de datos son obtenibles por el sistema adquisisión y procesables por la solución propuesta. Esto permite indicar que la solución tiene un nivel de intrusión relativamente nulo.

A continuación se explican todos los bloques de color gris y amarillo de la solución de manera extensiva, indicando las entradas y salidas de cada uno y que proceso y algorítmo usará.

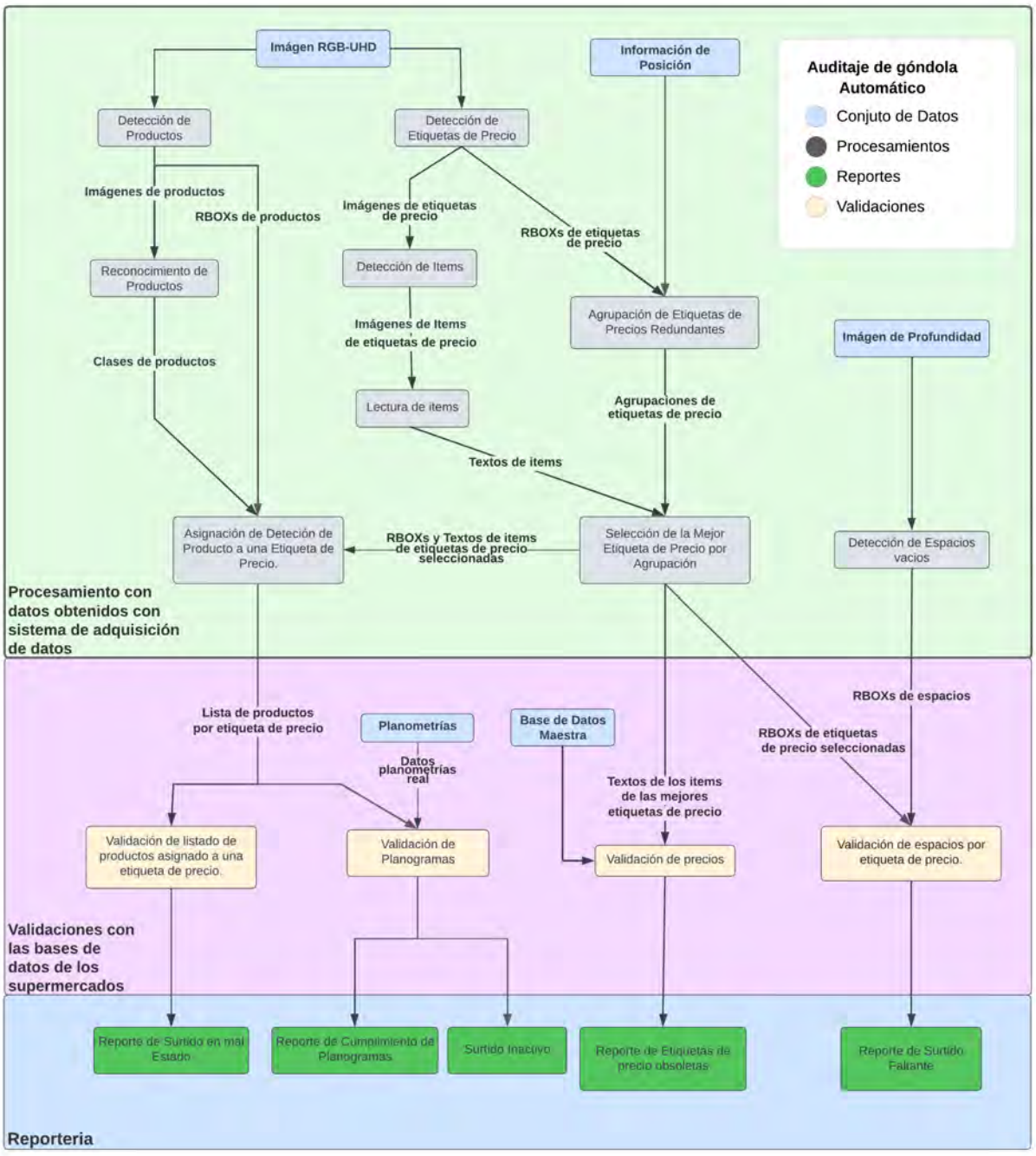

Figura 3.4: Propuesta de flujo de trabajo para resolver el problema de audita de góndolas.

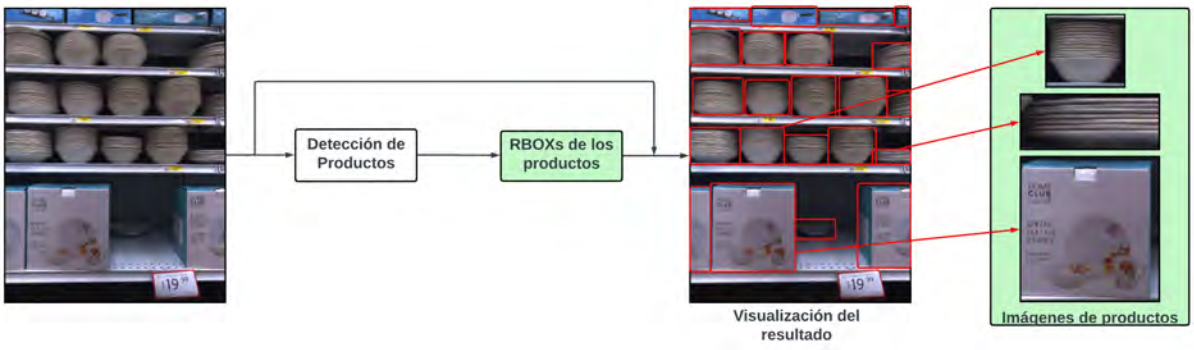

Figura 3.5: Visualización general del proceso de detección de productos.

3.3.1 Detección de productos

Este proceso utiliza las imágenes RGB-UHD (elemento del conjunto de datos obtenidos por el sistema de adquisición) como entrada y produce RBOXs de los productos detectados en la escena.

Cada RBOX está definido como una lista de 6 valores similar a $[X, Y, H, W, \alpha, CONF]$, donde:

- X representa el valor normalizado de la posición del centroide del RBOX en referencia a las columnas o anchura de la imágen.
- Y representa el valor normalizado de la posición del centroide del RBOX en referencia a las filas o altura de la imágen.
- H representa el valor normalizado de la altura del RBOX en referencia a la altura de la imágen.
- W representa el valor normalizado de la anchura del RBOX en referencia a la anchura de la imágen.
- α representa el ángulo de rotación del RBOX con respecto al eje de la imágen, valores entre (90°,-90°)
- $CONF$ representa el valor de confianza de la detección en que el objeto detectado es un objeto realmente.

Algunas implementaciones pueden incluir un séptimo valor en la lista, este es $CLASS$ que representa la clase del objeto detectado, sin embargo, este valor se vuelve trivial en este punto pues solo la tarea de detectar los productos en ambientes altamente densos es una tarea compleja, por tanto la clasificación o reconocimiento del producto se realiza en un bloque diferente para evitar sobrecarga en el modelo utilizado en el bloque.

Para este bloque se utilizará un algorítmo de detección de objetos con metodología de aprendizaje profundo usando redes neuronales convolucionales profundas y que será previamente entrenado para llevar a cabo el proceso de detectar los productos de la escena sin clasificarlos.

Finalmente, este bloque creará recortes de los objetos detectados utilizando los RBOXs para ser utilizados como parte de la entrada al siguiente bloque, y los cuales serán referidos como *images de productos*. Una ilustración del proceso se puede visualizar en la figura 3.5, donde también se visualizan los RBOXs de las detecciones dibujados en la imágen como recuadros de color rojo.

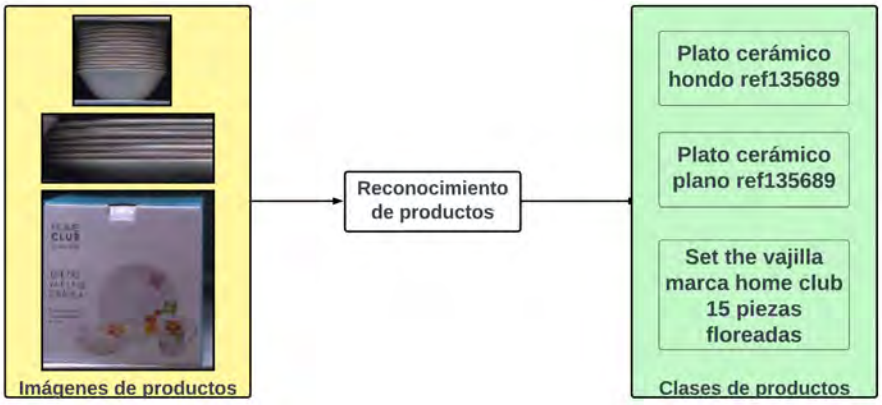

Figura 3.6: Visualización general del proceso de reconocimiento de productos.

3.3.2 Reconocimiento de productos

Este proceso utiliza las imágenes de productos como entrada y produce un texto que hace referencia a la clase de producto.

Este bloque es incluído en el flujo de manera separada al bloque de detección para evitar un sobrecargo en la tarea de detección de productos.

El texto resultante puede ser una descripción ó el código del producto, pero se recomienda usar los códigos en lugar de una descripción, ya que el conjunto de datos será más liviano; además, la probabilidad de cambiar la descripción de los productos ligeramente es superior a la de cambiar códigos. Al resultado de este bloque se lo denominará *clases de productos*.

Para este proceso se utilizará un algorítmo de aprendizaje de máquina guiado a la tarea de multiclasificación de objetos. Cabe destacar que esta tarea al día de hoy es un problema abierto a la comunidad científica ya que no existe una forma explicita y directa de solucionarla. En la figura 3.6 se puede visualizar el proceso general con ingresos y salidas.

3.3.3 Detección de espacios vacios.

Este proceso utiliza las imágenes de profundidad (elemento del conjunto de datos obtenidos por el sistema de adquisición) como entrada y genera RBOXs de los espacios encontrados en las góndolas de la tienda.

Para este proceso se puede utilizar la información de distancias entregada en las imágenes de profundidad, donde cada píxel de la imagen contendrá el valor cuantitativo en centímetros (u otra medida definida en el sistema de adquisición) de la distancia desde el plano paralelo al lente de la cámara 3D y la góndola. En la figura 3.7 se puede visualizar como la cámara ha obtenido las distancias y se puede reconstruir una imágen en escalas de grises (un solo canal) donde el valor de cada pixel representa la distancia medida desde el objeto hasta la cámara.

Al capturar las distancias, se puede utilizar un algorítmo de detección de relieves en los datos para encontrar aquellas zonas donde los relieves son pronunciados y de manera inversa (relieve ingresa hacia la góndola) y que se clasificarán como un espacio vacio.

Figura 3.7: Visualización del plano de la cámara 3D durante la captura de datos de distancia hacia los objetos de la góndola.

Para finalizar este proceso, el bloque tiene como salida los RBOXs definidos por las zonas de relieve pronunciado invertido y se denominarán *RBOXs de espacios vacíos.*

3.3.4 Detección de etiqueta de precio

Este proceso utiliza las imágenes RGB-UHD (elemento del conjunto de datos obtenidos por el sistema de adquisición) como entrada y produce RBOXs de las etiquetas de precios de los productos en la escena.

Similar al proceso de detección de productos, este proceso generará RBOXs como una lista, pero de 5 valores similar a $[X, Y, H, W, CONF]$, donde:

- X representa el valor normalizado de la posición del centroide del RBOX en referencia a las columnas o anchura de la imágen.
- Y representa el valor normalizado de la posición del centroide del RBOX en referencia a las filas o altura de la imágen.
- H representa el valor normalizado de la altura del RBOX en referencia a la altura de la imágen.
- W representa el valor normalizado de la anchura del RBOX en referencia a la anchura de la imágen.
- $CONF$ representa el valor de confianza de la detección en que el objeto detectado es un objeto realmente.

Para llevar a cabo este proceso, se usará un algoritmo de aprendizaje de máquina guiado a la tarea de detección de objetos y que será previamente entrenado para la tarea de detectar etiquetas de precios.

Particularmente para este bloque es necesario crear un conjunto de datos de entrenamiento para que el modelo pueda identificar correctamente cada diseño de las etiquetas de precio de el o los supermercados. Dado el poder de las redes neuronales convolucionales profundas, se estima que este conjunto no requerirá ser muy grande si se utiliza técnias de transferencia de aprendizaje.

Como paso final este bloque también realiza un recorte de las etiquetas de precios en la escena, las cuales se denominarán *imágenes de etiquetas de precio*.

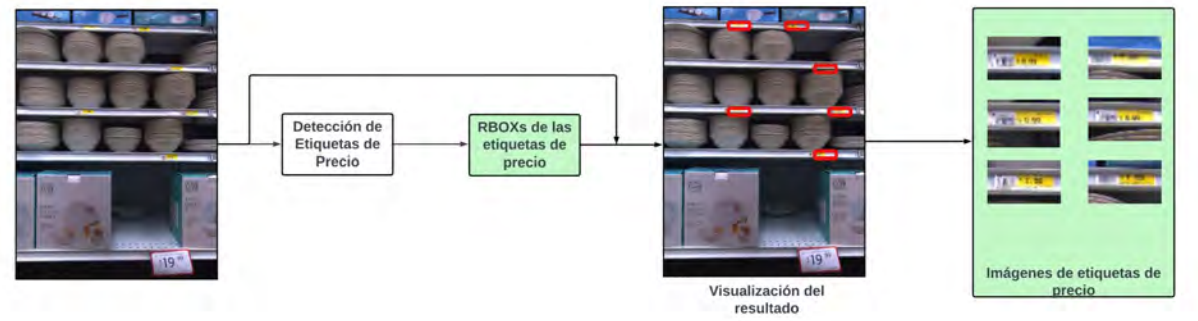

Figura 3.8: Visualización de la detección de etiquetas de precio en las imágenes RGB-UHD.

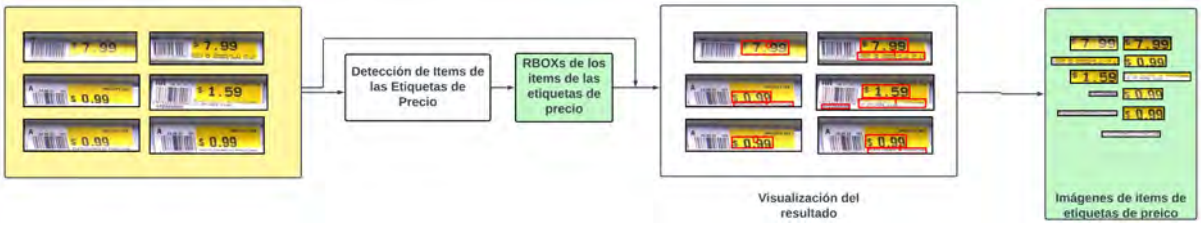

Figura 3.9: Visualización de la detección de los items de las etiquetas de precio en las imágenes RGB-UHD.

Es necesario hacer notar que el proceso de detección de productos y detección de etiquetas de precio pueden estar en un solo bloque y trabajarse bajo un solo modelo, sin embargo esto significaría realizar un conjunto de datos más extenso para que este modelo pueda entrenarse completamente. En la figura 3.8 se puede visualizar como se realiza este proceso y sus salidas. La detección de etiquetas es personalizada al modelo de etiqueta(s) de la tienda según el caso de uso.

3.3.5 Detección de items

Este proceso utiliza las imágenes de las etiquetas de precio como entrada y produce RBOXs de los *items de la etiqueta de precio*. Se considerarán como items de la etiqueta de precio a los textos que están presenten en la imagen de la etiqueta de precio; tendrán esta denominación para no confundir más adelante las interpretaciones con el reconocimiento de textos.

Estos items pueden ser, pero no están limitados a, el código del producto, precio del producto, descripción del producto y código de barras. En la figura 2.4a se puede observar un ejemplo de etiqueta de precio con sus items.

Para este bloque se utilizará un algortímo de aprendizaje de máquina guiado a la tarea de detección de objetos. No se propone un detector de textos pues como se indicó anteriormente uno de los items posibles de considerar puede ser el código de barras.

Para finalizar este bloque se generarán recortes de los items de la etiqueta de precios los cuales denominaremos *imágenes de los items de la etiqueta de precio.*. En la figura 3.9 se puede visualizar el proceso de la detección de los items de la etiqueta de precio. Este proceso puede dar como resultado diferentes cantidades de items por cada etiqueta de precio por el hecho de que las etiquetas puedes

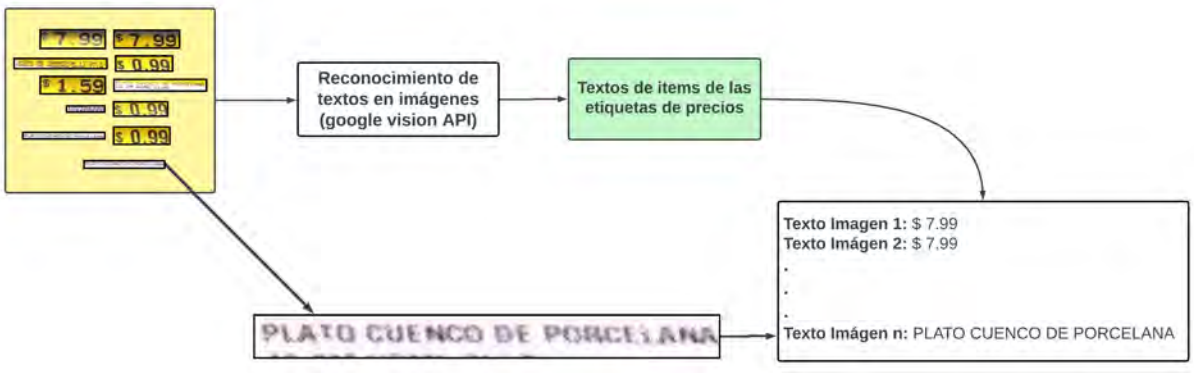

Figura 3.10: Visualización del proceso y resultado de lectura de items de etiquetas de precio.

estar ocluídas por el ángulo de una cámara, pero gracias al diseño de redundancia que tiene el sistema de adquisición, estos no presentan problemas y pueden ser identificados y eliminados posteriormente.

3.3.6 Lectura de items

Este proceso utiliza las imágenes de los items de la etiqueta de precio como entrada y produce textos de los items. Esto se debe aclarar pues, generalmente no se tendrán solamente textos, dado que los diseños de las etiquetas de precios es amplio. En este trabajo solo se utilizarán textos, pero se deja claro que es posible aumentar el bloque con otras formas de lectura como lo son el de código de barras (EAN-13).

Para la lectura de los textos de las imágenes de los items de las etiquetas de precio, se utilizará una metodología de reconocimiento óptico de caracteres (OCR por sus siglas en inglés). Para esto, y evitando la creación y entrenamiento de datos masivos para este bloque, se utilizó el API de la plataforma nube de Google (GCP, google cloud platform por sus siglas en inglés). El uso de esta plataforma es considerado momentaneamente hasta poder obtener un cojunto de datos etiquetados para profundizar en algorítmos de OCR y tener un modelo propio.

En la figura 3.10 se puede observar el uso de las imágenes de los items de las etiquetas de precio usando la API de google para realizar OCR y entregar los textos de estos. El proceso de configuración y uso de la API es simple y no se profundizará aquí pues se escapa del tema. Sin embargo, se pueden ver los resultados que se obtienen siendo muy buenos, y agregando que almacenando estos datos, se obtendrá en poco tiempo un conjunto de datos lo suficientemente grande para entrenar un modelo propio y evitar el consumo de la API.

3.3.7 Agrupación de etiquetas de precios redundantes

Este proceso utiliza múltiples entradas:

- **RBOXs de etiquetas de precio:** En el proceso de Detección de Etiquetas de precio, uno de los pasos es detectar las etiquetas en la escena y obtener los RBOXs de las mismas.
- **Información de posición:** Elemento obtenido por el sistema de adquisición de datos usando la localización del mismo en un sistema relativo a la tienda.

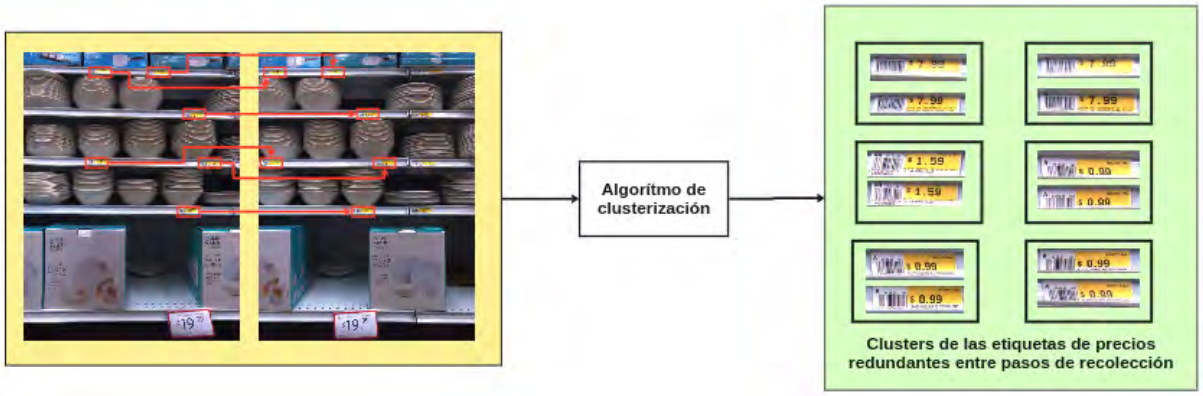

Figura 3.11: Visualización de resultado obtenido por la agrupación de etiquetas de precios redundantes.

- **Imagen de profundidad:** Elemento obtenido por el sistema de adquisición de datos usando la cámara 3D que cada Estructura de Colector poseé.

Todas las entradas juegan una parte importante en este bloque, el cual generará agrupaciones o clústeres de las etiquetas de precio. Para comprender exactamente este paso, se debe resaltar lo mencionado en la Sección 3.2, donde existirán zonas de interlape que se denominan *redundancias verticales* y *redundancias horizontales*. Gracias a estas redundancias no se pierde información durante la recolección, pero sobrecargaría el sistema. Para esto, el proceso pasa por varias etapas donde se ubicará cada etiqueta en un espacio bi-dimensional paralelo a la góndola, donde se podrá realizar agrupamiento de las etiquetas mediante su posición relativa a este espacio.

Esto será posible de logar haciendo una proyección del la posición de las etiquetas en la imagen RGB-UHD a imagen de profundidad, y mediante matrices de traslación y rotación ubicar en el mapa relativo al local usando la información posicional.

En la figura 3.11 se puede visualizar dos imágenes RGB-UHD de dos pasos consecutivos del sistema de adquisición de datos. Estas imágenes tienen espacios redundantes, y por tanto, también contienen etiquetas de precios redundantes. Como se indica en la figura, varias de las etiquetas se repiten, por tanto el algorítmo generaría una clusterización o agrupación de estas etiquetas mediante el parámetro de posición bi-dimensional que se obtenga. Es necesario destacar que en cada grupo de etiquetas redundantes existen etiquetas de mala, baja y alta calidad para los procesos posteriores, por lo que es necesario realizar los debidos filtros para obtener las mejores etiquetas posibles.

Para finalizar, este proceso dará como salida un listado de ids de etiquetas de precios agrupadas las cuales se denomirá *Agrupaciones de etiquetas de precio.*

3.3.8 Selección de la mejor etiqueta de precio por agrupación

Este proceso utiliza los Textos de items y Agrupaciones de etiquetas de precio. La principal razón de este bloque es poder segregar las agrupaciones y filtrar posible errores cometidos durante la agrupación usando solo la información posicional.

Los Textos de los items son utilizados en las etiquetas de cada agrupación para hacer una segrega-

Figura 3.12: Visualización de la selección de las etiquetas de precio por cada cluster generado escogiendo las mejor de todas. En caso de no haber alguna etiqueta que supere las cotas mínimas definidas, no se seleccionará etiqueta de ese cluster.

ción interna. Sabiendo que las etiquetas de cada agrupación son aquellas que están muy cercanas, es probable que dado el escenario denso con el que se está trabajando, en múltiples agrupaciones estén presentes redundancias de etiquetas de 2 ó más productos reales. Por esta razón se pueden utilizar los Textos para segregarlos usando algorítmos que permitán disociar cuales textos son más cercanos que otros y poder separar estas agrupaciones en 2 o más grupos.

Una vez que cada agrupación ha pasado por el filtro de segregación usando los textos de los items, se podría asegurar con gran probabilidad que en cada agrupación solo existen redundancias de un mismo objeto o etiqueta de precio. Así mismo, usando la confianza de lectura de los textos de los items, se puede generar una confianza ponderada según la relevancia de cada item. Consecuentemente, se pueden ordenar las redundancias ponderadas de etiquetas en cada agrupación y seleccionar aquella que tenga el mayor valor para cada agrupación.

Finalmente este proceso entregará como salida del bloque *RBOXs y textos de los items de las mejores etiquetas de precio.*

3.3.9 Asignación de deteción de producto a una etiqueta de precio.

Este proceso utiliza 4 entradas:

- **RBOXs de los productos:** Salida del bloque de Detección de Productos.
- **RBOXs de las etiquetas de precio seleccionadas:** Salida del bloque de Selección de la Mejor Etiqueta de Precio por Agrupación.
- **Clases de cada RBOXs de los productos:** Salida del bloque de Reconocimiento de Productos.
- **Texto de los items de las etiquetas de precio selecionadas:** Salida del bloque de Selección de la

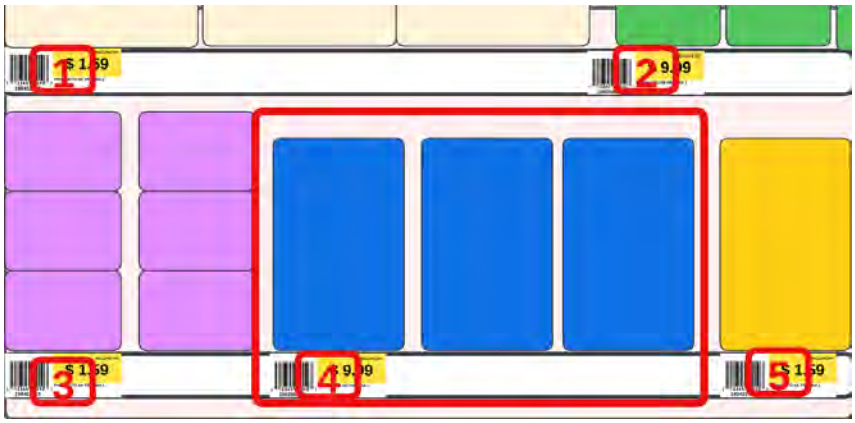

Figura 3.13: Posicionamiento de las etiquetas de precios en las góndolas.

Mejor Etiqueta de Precio por Agrupación.

Este proceso es uno de los principales para un correcto auditaje de góndola, pues los objetos reconocidos en la escena serán agrupados para dar un entendimiento general de lo que se está presentando realmente en las góndolas de las tiendas. Principalmente este bloque utiliza las etiquetas de precios seleccionadas (RBOXs) para organizar las detecciones de los productos. En la figura 3.13 Los productos de color azul están encerrados en un recuadro color rojo. Sobre este tipo de organización se enfocará este bloque. Para lograr esto se seguirán los siguientes pasos:

1. Agrupar todas las etiquetas según la bandeja a la que pertenezcan.

2. Enlistar las etiquetas de cada bandeja de izquierda a derecha usando la posición de la etiquetas (RBOXs)

3. Por cada bandeja definida, separar las detecciones de productos en grupos.

4. De izquierda a derecha se agrupará las detecciones de los productos con la etiquetas. Los limites de estas agrupaciones serán dados desde el borde inicial izquierda de la detección de una etiqueta de precio hasta el siguiente borde inicial izquierdo de la siguiente detección de etiqueta de precio en la misma bandeja.

Siguiendo estos pasos, la etiqueta de precio número 4 en la agrupación indicada con color rojo sera realizada mediante la limitación entre bandejas primero (No sobre pasa la bandeja en la cual se encuentran las etiquetas con número 1 y 2), y luego entre los bordes iniciales de la etiqueta número 4 y 5 como se puede apreciar en la figura 3.13.

Estas agrupaciones se realizarán con la finalizadad de poder auditar que productos están siendo puestos en las caras del producto indicado por la etiqueta de precio. Con esto, se puede reducir a identificar cada agrupación e internamente validar la consistencia de los productos y la etiqueta de precio.

Como salida para este bloque se tendrán *listas de productos por etiqueta de precio*.

3.3.10 Validación de precios

Este es el primer bloque de validación que se explicará y por tanto es necesario indicar que los bloques de validación son procesos que mayormente utilizarán las entradas para contrastar datos de las bases de datos del supermercado y finalizaran generando alertas que serán utilizadas por los bloques de reportería (color verde) para generar un reporte en formato PDF para que los operadores de la tienda puedan leerlo y resolver los problemas rápidamente.

Este bloque utiliza la salida del bloque de Selección de la mejor etiqueta de precio por agrupación: Textos de los items de las mejores etiquetas de precio. La validación se enfoca en contrastar los datos obtenidos en la tienda y los datos de la base datos para obtener cuales productos tienen un precio equivocado u obsoleto. En este particular, se debe tener almenos un campo que pueda emparejarse con los datos adquiridos y leídos durante el proceso y un campo obligatorio de precio para validar con los datos procesados. El resultado de este proceso será una lista de alertas de etiquetas de precio con el precio desactualizado (etiquetas de precios obsoletas) en las góndolas.

3.3.11 Validación de espacios por etiqueta de precio

Este bloque utiliza las salidas del bloque de Selección de la mejor etiqueta de precio por agrupación: RBOXs de las mejores etiquetas de precio, y del bloque de Detección de Espacios vacios: RBOXs de espacios. La validación se enfoca en emparejar los espacios vacios con una etiqueta estimando cual es el producto que falta por perchar en la góndola. Este proceso es similar al visto en el bloque de Asignación de detección de producto a una etiqueta de precio, pero con la variante de que del RBOX del espacios se encuentra que etiqueta es la más cercana bajo el espacio y a la izquierda. El resultado de este proceso será una lista de alertas de productos faltantes (surtido faltante) en las góndolas.

3.3.12 Validación de listado de productos asignado a una etiqueta de precio.

Este bloque utiliza la salida del bloque de Asignación de Detección de Producto a una Etiqueta de Precio: Lista de productos por etiqueta de precio. Este proceso se enfoca en contrastar el listado de productos por cada etiqueta de precio y encontrar discordancias producto-etiqueta, por ejemplo, si una de las listas de producto tiene como etiqueta al producto P1, y los productos listados son [P1,P1,P1,P2,P1,P1,P3], significa que esos productos P2 y P3 están mal ubicados ya que están en el frente de bandeja del producto P1. El resultado de este proceso será una lista de alertas de productos mal perchados (Surtido en mal estado) en las góndolas.

3.3.13 Validación de planogramas

Este bloque utiliza la salida del bloque de Asignación de Detección de Producto a una Etiqueta de Precio: Lista de productos por etiqueta de precio y el dato de Planometrías del conjunto de datos propuesto. Este proceso se enfoca en contrastar el listado de productos por cada etiqueta de precio y la planometría, encontrando un orden según las estructuras definidas para dar como resultado un valor cuantitativo del cumplimiento de planogramas del local. Esto es, cada góndola será calificada según su planograma para finalmente promediar la calificación de todas las góndolas y tener un valor de cumplimiento para la tienda. El resultado de este proceso será una calificación de cumplimiento de planogramas, que puede ser segregada desde local a sección o zonas según se requiera en el reporte. Un punto

importante es que con la misma información se puede sacar el complemento y saber que productos no han sido colocados en las góndolas y obtener otro dato importante que es el surtido inactivo.

4

Etiquetas de Precio Obsoletas

En este capítulo se aborda de manera más profunda una parte del flujo propuesto en el capítulo anterior, el cual entregará como resultado un reporte denominado etiquetas de precios obsoletos.

El reporte de etiquetas de precios obsoletos es uno de los reportes más importantes en la auditoria de góndolas. En este contexto, la etiqueta de precio destaca como uno de los objetos primordiales de la góndola, tal como ha sido mencionado previamente en este trabajo, y esto se debe a que es el principal comunicador del precio de los productos de las góndolas a los clientes. De hecho, se podría considerar como el canal de comunicación principal entre la góndola de la tienda y los clientes. Por esta razón, es de vital importancia mantener este elemento en el mejor estado posible y constantemente actualizado.

Cabe señalar que las etiquetas de precios vienen en diferentes formas y diseños. Normalmente son personalizados por cada negocio minorista para sus propias necesidades, un ejemplo de etiqueta de precio se muestra en la figura 2.4a. Las etiquetas de precio muestran textos e imágenes, que de aquí en adelante serán denominados *items*. Estos textos e imágenes pueden ser la descripción del producto y el código de barras; estos items brindan información sobre el producto a los clientes minoristas y empleados. A modo de ejemplo:

- **Código de barras:** ayuda a los operadores a reconocer rápidamente los productos mediante el uso de un escáner de código de barras incluido en un dispositivo para el inventario.
- **Texto del precio:** ayuda a los clientes a saber el valor que se pagará por el producto

Otros items pueden ser porcentaje de descuento, código de producto (normalmente un código interno usado solo por el minorista), QR (similar al código de barras pero generalmente puede recuperar más información). Es obvio inferir que cada etiqueta de precio puede tener diferentes cantidades y tipos de *items*, pero es claro que hay 3 *items* que siempre necesitan estar en una etiqueta de precio, estas son: Precio, Descripción y Código de Barras.

Para este trabajo se utilizará el diseño presentado en la figura 2.4a. Este diseño tiene los items Precio, Descripción, Código de Producto y Código de Barras. Singularmente el uso de un único diseño podría

ser visto como un problema de generalización, sin embargo se preveé que con el uso de los algorítmos de inteligencia artificial como lo son las redes neuronales convolucionales profundas, se tendrá la facilidad de generalización con el aumento de datos en el conjunto de datos. Lo importante en definir son los pasos efectivos y coordinados para la elaboración del flujo que los datos deberán seguir para que se pueda extraer la información y ser validados. Para lograr esto, en este capítulo se presentará a detalle parte del flujo de trabajo mencionado en el capítulo anterior con enfoque a obtener el reporte de etiquetas de precios obsoletas.

Primero, se profundizará en la metódología de adquisición de los datos y sus pros y contras, luego, se presentarán ejemplos del subconjunto de datos usado para esta parte del flujo. A continuación, se presentará la mecánica de trabajo en los bloques presentes para esta parte del flujo. Finalmente, se presentarán resultados de los bloques con métricas selecionadas para medir el rendimiento de los algorítmos utilizados.

4.1 Flujo de adquisición de los datos

En la sección 3.2 se presentó el diseño del sistema de adquisición de datos (robot). Este robot realiza recorridos autónomos en la tienda del supermercado. El robot fue programado utilizando el software ROS (Robot Operating System), mismo que es utilizado mundialmente como base para desarrollo robótico. ROS tiene compatibilidad nativa con Python3 desde la versión NOETIC, misma que se utilizó para el desarrollo del robot. Durante los recorridos, el robot realiza dos tipos de movimiento:

- **Movimiento libre:** El robot utiliza el algortítmo de planificación de movimiento en ROS para moverse de un punto a otro evadiendo obstáculos.
- **Movimiento Seguidor de Góndola:** El robot utiliza las cámaras 3D para medir la distancia entre sí mismo y la góndola manteniendo una linea recta desde el inicio hasta el final del movimiento.

A modo general, el robot se mueve iterando entre estos dos movimientos. A modo de ejemplo, el robot se moverá de forma libre desde cualquier punto hasta el inicio de una cabecera de góndola o inicio de pasillo, desde este punto el robot se moverá en forma de Seguidor de Góndola, manteniéndose a una distancia constante de la góndola. Este tipo de movimiento realiza *pasos* de 25 cm cada uno. Luego de cada paso, el robot se detiene y captura información (captura de imágenes de modo asíncrono en cada colector). Y así continúa hasta llegar a la meta, que es el final del pasillo o la cabecera de góndola opuesta. En la figura 4.1 se puede observa una ilustración de estos dos movimientos que realiza el sistema de adquisición, las flechas de color verde indican un movimiento libre y por tanto los trazos no son rectos, mientras que las flechas color azul indican un movimiento seguidor de góndola, el cual se realiza paralelo a la linea de la góndola y siempre tratando de ser recto como se ilustra. La combinación de estos dos movimientos permite al sistema de adquisición de datos moverse en el área de venta y recolectar toda la información necesaria. Así mismo, si existen obstaculos el en trayecto de un movimiento de seguidor de góndola, el sistema de adquisición pasa a un movimiento de tipo libre para evadir el obstaculo y colocarse en el siguiente punto posible de continuar el movimiento de tipo seguidor de góndola.

Las cámaras de la estructura de colección son acopladas con la unidad de procesamiento en una estructura personalizada diseñada a medida para que estén lo más cerca posible, intentando que apunten a la misma escena. Está claro que la escena nunca será exactamente la misma pues no es posible montar

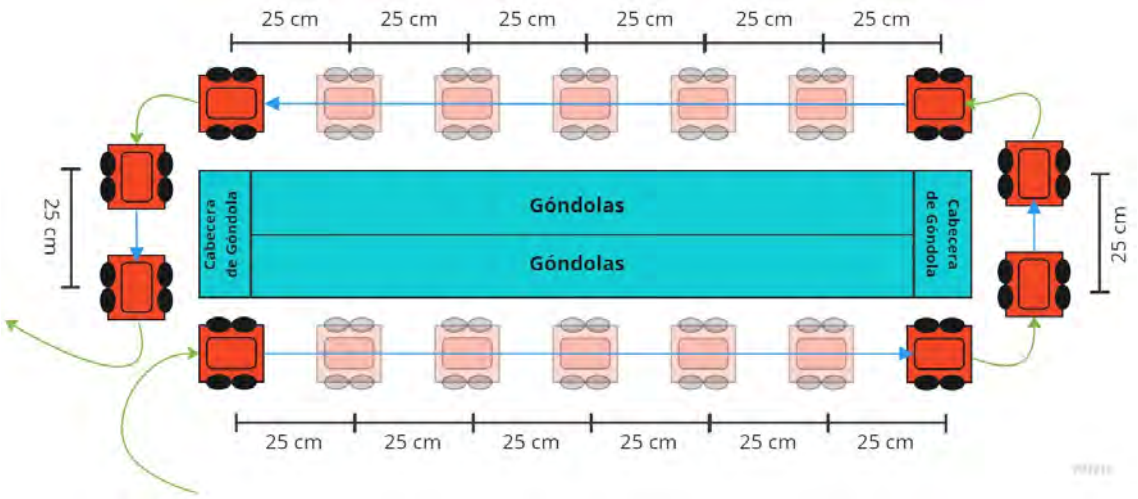

Figura 4.1: Movimiento del sistema de adquisición de datos. Tipo de movimiento libre en color verde y tipo de movimiento seguidor de góndola en color azul. Cada paso del robot es de 25 cm aproximadamente.

los sensores sobre el mismo espacio físico, pero dado su gran cercanía se puede inferir que las escenas observadas por ambas cámara serán casi iguales. Para esto, también hay que considerar los diferentes campos de visión de cada cámara y que más adelante serán oportunamente trabajados para que se pueda proyectar los pixeles de una imagen sobre la otra usando un proceso de calibración entre las cámaras.

Dada la naturaleza de la colección del conjunto de datos y la posición de las cámaras, habrá espacios de góndola repetidos. En la figura 4.2 (Pase 1 y Paso 2) se puede apreciar la redundancia entre los colectores en los recuadros amarillos (redundancia vertical), así como la redundancia entre los pasos del robot en los recuadros morados (redundancia horizontal). Estas redundancias evitan capturas parciales de las góndolas. Esto también genera retomas de los objetos, es decir, un mismo objeto puede ser capturado dos o más veces, por diferentes cámaras y en diferentes pasos. Esto indica que existirá redundancia de etiquetas de precio, lo que favorece que el proceso sea presentado posteriormente, ya que existirán varias *instancias* de un mismo objeto, de las cuales se podrá elegir la mejor y se podrá validar mejor la información.

4.2 Conjunto de datos

Los tipos de datos a usarse para el desarrollo de esta sección son:

- **Imágenes RGB-UHD:** Imágenes típicas RED-GREEN-BLUE recopiladas con las cámaras UHD o de ultra alta definición. La resolución de estas imágenes es de 3684x4912 píxeles. Cámara de la marca IDS, modelo UI-3590CP Rev2 con Lente KOWA montura tipo C modelo LM5JC10M, (obtenido por cada colector, Figura 3.3: cámaras RGB-UHD superior, medio e inferior).
- **Imágenes de profundidad:** Imágenes de tipo 3D obtenidas con cámaras 3D. Este tipo de cámara usa láser para medir la distancia desde la lente hasta los objetos en la escena. Por lo tanto, cada valor de píxel representa la distancia medida. La resolución de estas imágenes es de 480x640

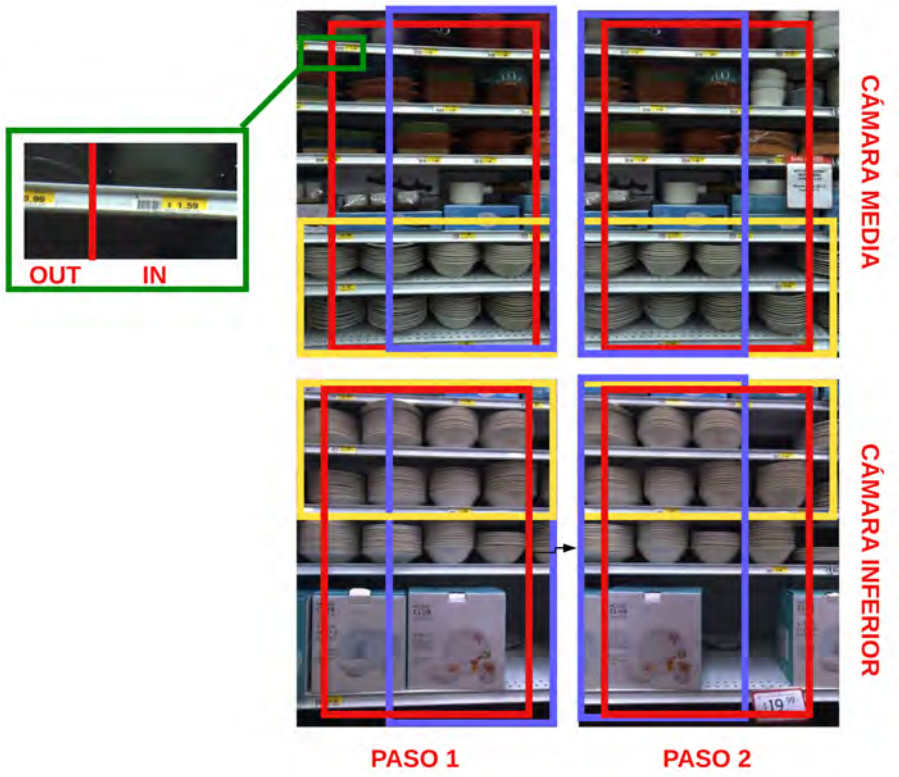

Figura 4.2: Imágenes RGB-UHD obtenidos de dos colectores (imágenes en vertical) y en dos pasos consecutivos (imágenes en horizontal) del robot. Redundancia vertical observada con recuadros amarillos entre colectores, y Redundancia horizontal observada con recuadros morados entre los pasos (steps). En verde se hace zoom de una etiqueta de precio parcial (etiqueta izquierda) y que se elimina con ayuda del filtro de color rojo en cada imágen. Los dos colectores usados del robot son: cámara RGB-UHD media y cámara RGB-UHD inferior.

píxeles. Cámara de la marca OBBEC, modelo Astra, (obtenido por cada colector, Figura 3.3: cámaras 3D superior, medio e inferior).

- **Información de posición:** Coordenadas X, Y y el ángulo YAW del robot en relación al mapa de la tienda, (obtenido mediante una mezcla de sensores: Lidar y la cámara 3D de navegación, Figura 3.3).

- **Base de Datos Maestra:** Una base de datos para consultar información de los productos como precios, descripciones, códigos de barra, entre otros.

Cabe destacar que de estos 4 tipos de datos, 3 son recolectados por el sistema de adquisición mencionado anteriormente. El tipo de dato **Base de Datos Maestra** se recupera de las bases de datos del supermercado.

Una sola ejecución del proceso de adquisición del robot proporciona una gran cantidad de datos, por lo que para este trabajo se separaron dos conjuntos de datos.

Tabla 4.1: Cantidades de imágenes por conjunto de datos de imágenes para entrenamiento de detección de etiquetas de precio e items de las etiquetas de precio.

	Detección etiquetas	Detección items
Entrenamiento	632	2087
Validación	154	853
Prueba	65	541
Total	851	3481

4.2.1 Conjunto de datos para entrenamiento de modelos

Este conjunto de datos es utilizado para entrenar modelos de detección de objetos y se separa en sub-conjuntos de entrenamiento, validación y prueba para el típico flujo de entrenamiento y prueba de un modelo de detección de objetos.

Se compone únicamente de imágenes RGB-UHD para entrenar modelos de detección de objetos, un ejemplo puede ser observado en la imagen que se muestra en el lado izquierdo de la figura 3.2, además son recopiladas aleatoriamente de diferentes recolectores en diferentes pasos del robot. Este conjunto de datos se usa solo para entrenar los algoritmos de detección de objetos en el subflujo de trabajo para la obtención del reporte de etiquetas de precios obsoletas, por esta razón las imágenes son aleatoriamente escogidas, permitiendo más generalidad a los modelos entrenados y evitar sesgos producidos por tener imágenes de capturas continuas.

Se enfatiza que estos conjuntos de datos consisten en etiquetas de precios con un solo diseño específico. Por lo cual, el modelo entrenado en esta sección solamente se centra en la detección de dicho diseño. Para poder aplicar la solución propuesta en un ambiente con distintos diseños de etiquetas de precios, se entrenaría el modelo con un dataset diverso que contenga los diseños deseados. Después de ser entrenado en el nuevo conjunto de datos, el modelo debería poder detectar etiquetas de precio de ese nuevo diseño con alta precisión, manteniendo intacto el resto del flujo del trabajo propuesto.

El conjunto de datos de entrenamiento fue etiquetado manualmente para detectar las etiquetas de precios de los productos en las góndolas. Luego, los RBOX generados durante este etiquetado manual también fueron reunidos en otro conjunto de datos para realizar detección de items en las etiquetas de precios.

En el cuadro 4.1 se pueden ver la cantidad de datos para el entrenamiento de detección de etiquetas de precios y de items de las etiquetas de precio. En este punto debe aclararse que el conjunto de datos para entrenamiento de modelos debe dividirse en los 3 típicos sub-conjuntos: de entrenamiento, de validación y de prueba.

4.2.2 Conjunto de datos para prueba completa

Este conjunto de datos es utilizado para evaluar los modelos entrenados, los pasos de agrupación del modelo y la reportería final.

Tiene un lado de dos pasillos diferentes, llamados CERO y UNO. La figura 3.2 muestra un ejemplo de imágenes adquiridas en el pasillo UNO. La cantidad de imágenes se menciona en el cuadro 4.2, aquí los pasos significan la cantidad de paradas que realizó el robot durante la recopilación de imágenes en

Tabla 4.2: Cantidades de imágenes por conjunto de datos para pruebas completas. Son dos pasillo "CERO" y "UNO".

	Pasillo CERO	Pasillo UNO
PASOS (Robot)	16	29
Ubicaciones	16	29
Imágenes RGB	48	87
Imágenes de profundidad	48	87

dicho pasillo, por lo tanto, más pasos significan un pasillo más largo. Para cada paso también se guarda la información posicional. Las imágenes RGB-UHD y de profundidad son tres veces el número de pasos porque el robot tiene tres colectores (superior, medio e inferior).

Este conjunto fue etiquetado de forma manualmente, con la finalidad de poder ser validado tanto para los modelos entrenados para detectar etiquetas precio e items de etiquetas de precio. Así mismo, este conjunto aumenta más datos y por ende el etiquetado es más extenso. Luego de tener los items etiquetados, se procedió a manualmente escribir los textos de cada item y guardarlos para poder evaluar el procesamiento de reconocimiento de textos. Así mismo, para la localización y reducción de redundancias de etiquetas (este proceso se explica a detalle más adelante) se elaboró una agrupación de etiquetas de precio de manera manual de tal forma que en cada grupo existan solo redundancias de imágenes pertenecientes a la misma etiqueta de precio real.

4.2.3 Consideraciones extras

En el caso del diseño de la etiqueta de precio presentado en la figura 2.4a, solo se utilizarán dos de los ítems visibles, estos son: Precio y Descripción. Aunque el código de barras es uno de los elementos principales de una etiqueta de precio y está presente en este conjunto de datos, no se utilizará, principalmente porque el código de barras de cada etiqueta de precio tiene una dimensión pequeña debido a la forma en que se obtiene el conjunto de datos. Además que es problemático leer las líneas del código de barras con cámaras [39, 40]. Sin embargo, se puede implementar ya que existen enfoques de este tipo [41] pero considerando que requiere más computo o incluso hardware especializado no será utilizado en este trabajo.

4.3 Solución

A continuación se mostrará un diseño de flujo para solucionar el problema de Etiquetas de Precio Obsoletas. La solución propuesta se puede ver en la figura 4.3. Es importante tener en cuenta que esta solución se desarrolló en base a los datos recopilados. Sin embargo, se darán recomendaciones sobre qué bloques se pueden ampliar para mejorar la solución para diversos datos.

Las entradas para esta solución se pueden visualizar con bloques color naranja y como ejemplos de las imágenes (RGB-UHD y de profundidad).

El resultado del enfoque propuesto (bloque verde) es un informe que alerta al supermercado de las etiquetas de precios en el área de ventas que muestran precios obsoletos o desactualizados.

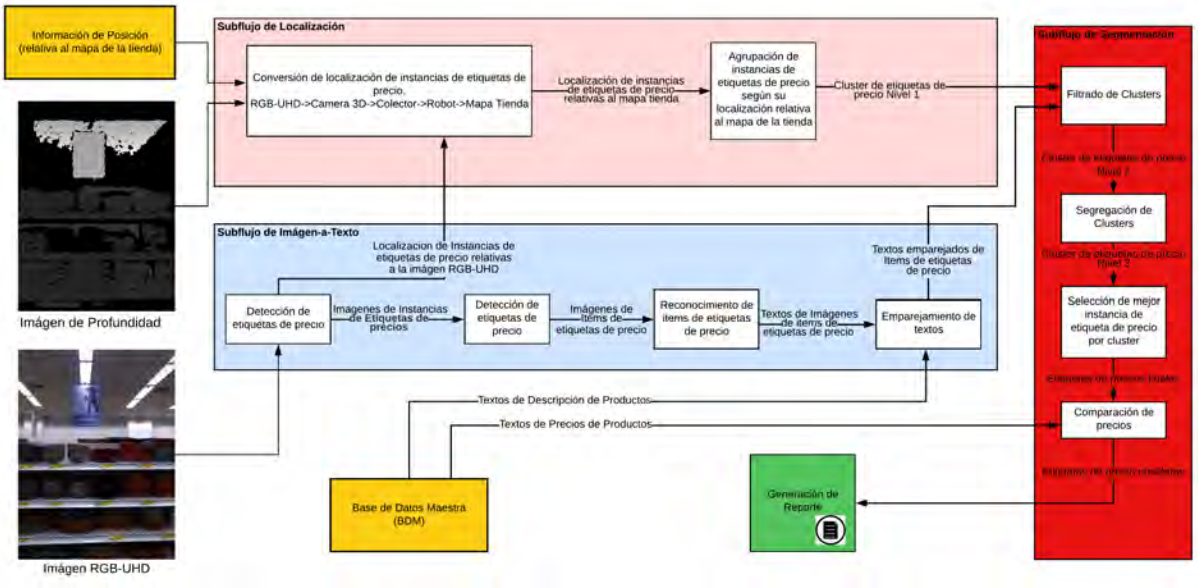

Figura 4.3: Flujo de trabajo para solucionar el problema de etiquetas de precio obsoletas. Incluye 3 subflujos de: Localización, Imagen-a-Texto y Selección.

En la figura 4.3 también están bien marcados los 3 subflujos antes mencionados. El subflujo de Imagen-a-Texto (color celeste), de localización (color rosado) y de selección (color rojo). Estos subflujos serán descritos a detalle a continuación.

4.3.1 Subflujo de imagen-a-texto

Este subflujo tiene como entradas las *imágenes RGB-UHD* recolectadas por el sistema de adquisición de datos (colectores superior, medio e inferior del robot), y los *textos de descripciones de productos* provenientes de la base de datos maestra (BDM). Este subflujo utiliza algoritmos de aprendizaje profundo y visión artificial para detectar, extraer y leer las etiquetas de precio de todas las imágenes RGB-UHD. Luego, utiliza las descripciones de productos de la BDM para comparar y emparejar los textos leídos con las descripciones de productos (que pueden contener errores menores). Este subflujo tiene dos salidas. La primera es la *localización de las instancias de las etiquetas de precio relativas a la imagen RGB-UHD* (usada como entrada por subflujo de Localización), y la segunda salida son los *textos emparejados de los items de etiqueta de precio* (usada por la pista de Selección como entrada). Cada bloque de esta pista se explicará con más detalle a continuación.

Detección de etiquetas de precio: Este es el primer bloque del subflujo de Imagen-a-texto. Utiliza las imágenes RGB-UHD como entrada y es responsable de detectar la ubicación de las etiquetas de precios en la imágen. Este bloque utiliza un modelo de inteligencia artificial dirigido a la detección de objetos con metodología de redes neuronales convolucionales profundas. Este modelo es YoloV5 [42] y se lo entrenó utilizando el conjunto de datos para entrenamiento. Usando técnicas de transferencia de aprendizaje se entrenó el modelo con 632 imágenes y validadas con 154 imágenes RGB-UHD previamente etiquetadas para la tarea de detección de objetos (el objeto es la etiqueta de precio).

En el cuadro 4.3 se puede observar los resultados de la Precisión Promedio (AP o average precision, por sus siglas en inglés) y aciertos (o hits) que obtuvo el modelo durante su entrenamiento para el subcon-

junto de prueba del conjunto de entrenamiento de modelo (consulte la Sección 4.2.1). Se debe hacer hincapie en que la métrica de 100% de aciertos significa que los n objetos en la escena fueron detectados y esto para todos los ejemplos del conjunto de datos; así mismo el AP o promedio de precisión, es calculado usando el IoU o Intersección sobre Unión de los RBOXs detectados, es decir que tan iguales llegan a ser.

Las detecciones de este modelo se entregan en el formato de: [x1, y1, x2, y2, confianza, clase], mismas que serán llamadas *Instancias de Etiquetas de Precio o IEP*. En la figura 4.3 se pueden observar 2 salidas del bloque de detección de etiquetas de precio, estas son explicadas a continuación:

- **Localización de IEPs relativas a la imagen RGB-UHD:** Esta salida corresponde a los elementos x1, y1, x2, y2 del formato de las detecciones.
- **Imágenes de IEPs:** Esta salida son recortes de la imagen RGB-UHD en los puntos (x1,y1) hasta (x2,y2). Este recorte es de la etiqueta de precio.

El modelo es capaz de detectar etiquetas de precios incluso cuando son parcialmente visibles en la imagen, normalmente ubicadas en los bordes de la imagen. Una instancia de una etiqueta de precio parcial se muestra en la figura 4.2, donde la etiqueta de precio izquierda (ampliada dentro del cuadro verde) está cortada por la mitad. Estos objetos parciales detectados no tienen ningún propósito práctico y, de hecho, suponen un obstáculo, ya que la información incompleta puede dar lugar a errores. Para abordar esto, se desarrolló un filtro para eliminarlos utilizando el centroide de la IEP y determinando si se encuentra dentro de un rectángulo interno (indicado en rojo) en la imagen RGB-UHD. Un ejemplo de este filtro es mostrado en la figura 4.2.

Las IEPs que pasen el filtro seguirán en el proceso del subflujo. Las IEPs se utilizarán para recortar la imagen RGB-UHD, obteniendo imágenes de instancias de etiquetas de precios (imágenes IEP), utilizadas como entrada del siguiente bloque. Asimismo, la ubicación de cada IEP, denominada Localización de Instancias de Etiquetas de Precio (Localización IEP), se utiliza como entrada para el subflujo de Localización.

Tabla 4.3: Resultados del promedio de precisión (average precision, AP) y aciertos (o hits) de detecciones de las etiquetas de precios y de los items de las etiquetas de precios.

	Detección de etiqueta de precio	Detección de Item de etiqueta de precio	
		Precio	Description
Precisión Promedio	96.43	95.17	91.78
Aciertos de Detección	100	99.89	97.93

Detección de items de etiquetas de precio: Este bloque utiliza la salida del bloque Detección de etiquetas de precio Imágenes de IEPs como entrada. Similar al anterior, este bloque usa el mismo modelo de red neuronal convolucional (YoloV5) para producir detecciones, pero con la diferencia de que los objetos ahora son los *items* de la etiqueta de precio. El formato continúa siendo el mismo [x1, y1, x2, y2, confianza, clase] pero ahora el elemento *clase* diferencia las clases de items que utiliza el proceso, como: Precio y Descripción.

El modelo YoloV5 fue entrenado con los datos extraídos del conjunto de datos de entrenamiento (con-

(a) (b) (c) (d) (e)

Figura 4.4: Ejemplos de Instanticas de Etiquetas de Precios con sus items.

sulte la Sección 4.2.1), el cual comprende 2087 imágenes de entrenamiento y 853 imágenes de validación. Los resultados de confianza de este modelo sobre el subconjunto de prueba (541 imágenes IEP) se pueden ver en el cuadro 4.3. Un dato destacable es que dado el diseño presentado en la figura 2.4a el item *Precio* se ven claramente en las Imágenes de IEP, mientras que el item *Descripciones* tienden a tener menos visibilidad dado que el texto es más pequeño y en el escenario real tendrá más probabilidades de no estar completo.

En la figura 4.4 se puede notar lo antes mencionado. Las Figuras 4.4e y 4.4d son ejemplos de IEPS donde los items se aprecian sin problema alguno; en las figuras 4.4c y 4.4b el item Descripción de producto, esta un poco menos visible; y, finalmente en la figura 4.4a está ya empezando a ser ocluído. Estos casos ocurren por la escena y por el diseño de la etiqueta de precio.

También hay que recordar que los colectores tienen espacios redundantes verticalmente, estos espacios suelen tener la mayor cantidad de IEPs con fallas de este tipo en las periferias superiores debido al FOV o campo de visión de la imágen. Se puede apreciar que los *códigos de producto* también podrían ser usados, pero en muchos más casos que descripción serán poco visibles, y a diferencia de la descripción, el código para poder ser emparejado requiere ser exacto, sino se podría emparejar con otro producto de manera errada.

En el cuadro 4.3 se encuentran los valores de Precisión Promedio y Aciertos para el modelo entrenado. Se hace notar que aunque es pequeña, existe una diferencia en estas métricas expresamente para el item Descripción y que se debe a los puntos antes mencionados.

En una Imagen IEP, solo puede haber una instancia de cada item, es decir, un Precio y una Descripción. Sin embargo, hubo casos en los que se detectó más de una instancia del mismo objeto en la Imagen IEP, es decir, dos o más detecciones de precios o descripciones en la misma etiqueta de precio. Para este caso, considerando que las detecciones tienden a superponerse, se realiza un filtro máximo simple sobre la confianza de las instancias. Al final, por cada Imagen de IEP habrá solo una instancia de cada objeto mencionado.

Las detecciones que pasen el filtro se llamarán Instancias de Items de Etiquetas de Precio (IIEP). Las detecciones se utilizan para crear imágenes recortadas denominadas Imágenes de Instancias de Items de etiquetas de precio (imágenes de IIEP) y se utilizarán como entrada para el siguiente bloque del subflujo de Imagen-a-texto.

Reconocimiento de items de etiquetas de precio: Este bloque utiliza como entrada la salida del bloque anterior: Imágenes de IIEPs. Esta parte del subflujo no se entrenó, en su lugar se utilizó una API [43], ya que actualmente se considera una tarea muy laboriosa entrenar un modelo para realizar el reconocimiento óptico de caracteres (OCR) en estos datos específicos. Los resultados obtenidos de la confianza de reconocimiento de esta API se registraron en el cuadro 4.4. Es necesario mencionar que esta API puede ser usada para tener resultados rápidamente y recopilar datos para generar un conjunto

Tabla 4.4: Estadísticas del reconocimiento de items de instancias de etiquetas de precios.

Reconocimiento de Etiqueta de precio	Confianza			
	Promedio	Desviación Estándar	Mínima	Máxima
Precio	95.44	4.24	64.02	99.42
Descripción	88.19	9.70	31.04	97.56

de datos lo suficientemente grande para entrenar un modelo sólido, a fin de eliminar la necesidad de pagar por el uso de la API.

Este bloque solo contiene el algoritmo OCR debido a la naturaleza de los items que se están utilizando (Precio y Descripción) por ser textos, sin embargo, podría contener otros algoritmos, como lectores QR o lectores de código de barras tipo EAN 13, para poder procesar otros items en la imagen de ser necesario.

Las respuestas entregadas por el API (Google Vision API) pueden contener errores. Esto es normal y debe ser premeditado. Entre los errores que se pueden dar son fallos o intercambios de letras en la lectura (una falla muy común) lo cual va a ser arreglado posteriormente. Otro forma de error del API es entregar textos con caracteres no alfanuméricos, para este caso se utiliza un filtro que elimina estos caracteres de los textos, evitando errores.

Los textos de salida de este bloque se denominarán Textos de Imágenes de Instancias de Items de Etiquetas de Precio (Textos de Imágenes de IIEP) y se utilizarán como entrada para el siguiente bloque.

Emparejamiento de etiquetas de precio con productos: Este bloque utiliza como entrada la salida del bloque anterior: Textos de Imágenes IIEPs.

Este bloque es uno de los más importantes y similar al bloque de reconocimiento de items de etiquetas de precio, es personalizable. La idea principal de este bloque es poder realizar un emparejamiento entre la etiqueta de precio y un producto de la base de datos. Para esto, se puede utilizar uno o múltiples items de la etiqueta de precio, por esta razón se dice que este bloque es personalizable. No hay una lógica única para este bloque y debe ser implementado de forma modular para poder agregar o desagregar condiciones de emparejamiento entre los actores. Los items de las etiquetas de precio seleccionados para este trabajo son Precio y Descripción. Por tanto, uno o todos estos items deben de ser usados en este bloque para lograr un emparejamiento robusto. En el caso de fallar en este bloque, se arrastrará un error considerable a lo largo del flujo completo.

En este trabajo, se utilizará solamente el item de Descripción. Este item tiene la particularidad de ser un texto que incluye solo dígitos alfanuméricos con exceptión de los caracteres *"/"* y *"-"*. Dado que la descripción puede ser corta o extensa, el uso de algorítmos como distancia de Levenstein dificultarían el emparejamiento considerablemente en las descripciones largas, pues miden los cambios de caracteres y al ser descripciones largas tienen más probabilidades de cambiar algunos caracteres. Sabiendo esto, se definió utilizar la distancia de cosenos entre textos con la finalidad de proyectar el texto reconocido en las imágenes y emparejarlo con una única descripción de producto de la base de datos. Los textos generados durante el reconocimiento de textos pueden contener pequeños errores, por ejemplo: *CAMISO BLANCA TAMANO XL* y *CAMISA BLANCA TAMANO XL*. En estos casos la distancia de cosenos

entre textos proporciona un valor numérico para medir la distancia entre la representación vectorial de los textos (si la distancia es 0 los textos comparados son exactamente iguales). Para la elaboración de esta representación vectorial se escogió un mapa de vectores generado con todos los posible 3-gramas de caracteres.

El método de emparejamiento es simple, se crea una matriz donde las filas están representadas por los Textos de las Imágenes IIEPs, y las columnas por los productos en el BDM. La matriz se llena con todas las distancias de Textos de las Imágenes IIEPs vs. Productos BDM. Luego, para cada fila, se elige la columna que tiene el valor más bajo para emparejar el texto generado con una descripción del producto del BDM. En esta parte, se está utilizando la coincidencia top 1, sin embargo, se podría hacer un top n de productos emparejados para llevar a cabo un análisis más profundo, talvez usando otros items. Posteriormente, se debe usar un umbral de distancia de coseno para evitar el uso de una distancia de coseno grande. Para este proyecto, usamos 0.001 como umbral, lo que significa que si la distancia de coseno más baja obtenida para el texto es más alta que el umbral, no se realizará ninguna coincidencia, ya que este texto podría haberse emparejado por error con un producto similar al real.

En lugar de crear una sola matriz, se recomienda generar varias matrices, cada una de las cuales se centra en productos que pertenecen a la misma categoría o alguna otra segmentación del supermercado. Principalmente una que divida a los productos en un solo pasillo, así se evitarían casos en que productos de categorias completamente distintas se emparejen. Evitando así por ejemplo, que porductos como detergente se asocien por error con productos de otra categoría, como las bebidas.

Los textos de salida siempre serán textos de descripción producto de la BDM que se denominarán Textos emparejados de Instancias de Items de etiquetas de precio (Textos Emparejados IIEPs). La salida de este bloque es también la salida de todo el subflujo de Imagen-a-Texto, misma que será usada en conjunto con la salida del subflujo de Localización como entradas para el subflujo de Selección.

4.3.2 Subflujo de localización

Este subflujo consta de tres entradas: Las Imágenes de profundidad, la información de posición (relativa al mapa de la tienda) y la localización de la IEPs (relativa a la imagen RGB-UHD). El propósito de este subflujo es generar grupos o clústeres de etiquetas de precios, en los que cada grupo contenga múltiples IEPs agrupadas por su cercanía espacial relativa al mapa de la tienda. La salida de este subflujo se denominará Clúster de Etiquetas de Precio Nivel 1 o CEPN 1. Como sugiere el nombre, representa el nivel inicial de agrupación realizado con a las IEPs en un sistema de coordenadas relativo a la tienda. Este nivel sirve como una primera aproximación y puede contener varios errores de agrupación, ya que la ubicación generada para cada IEP es una estimación. Estos posible errores luego son solventados en el subflujo de selección.

Conversión de localización de instancias de etiquetas de precio: Este bloque es uno de los más técnicos en relación al método de adquisición. Este bloque se relaciona con el sistema de adquisición de datos por utilizar la información de posición. Este bloque recibe todas las entradas del subflujo y se encarga de realizar las primeras transformaciones de las localizaciones relativas que tiene cada IEPs. Cada transformación realizada tiene una razón y se realiza mediante paquetes propios de ROS [44].

La figura 4.5 proporciona una representación visual de las transformaciones entre los diferentes sistemas de referencia involucrados. Cada flecha oscura representa un tipo de transformación. En el gráfico

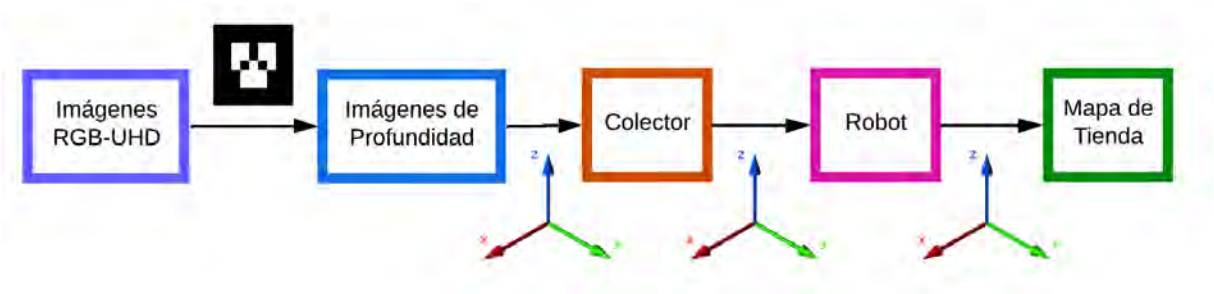

Figura 4.5: Conversiones de sistemas de referencias realizadas en el subflujo de Localización.

se podrán notar dos tipos de transformaciones: una representada por un ARTAG y la otra por un plano cartesiano.

El ARTAG representan una transformación o proyección entre las imágenes RGB-UHD y las imágenes de profundidad 3D. Ambas imágenes son capturadas por cada uno de los colectores del robot (colectores superior, medio e inferior, Figura 3.3). La estructura de cada colector está diseñada para que las cámaras RGB-UHD y 3D no se muevan, por lo que se puede suponer que están estáticas y siempre a la misma distancia. Sabiendo esto, se utilizaron algunos ARTAG para resolver automáticamente la proyección de los FOVs en los ejes X, Y [45] de la imagen RGB-UHD a la imagen de profundidad 3D. Dado que la estructura diseñada para contener los colectores es la misma para todos y las cámaras son del mismo modelo, solo se necesita una ejecución, lo que permite aproximar un píxel de la imagen RGB-UHD en un píxel de la imagen de profundidad 3D.

Los Planos Cartesianos representan una transformación o proyección utilizando la posición de los objetos internos del robot (cámaras, lidar, etc.). Este tema profundiza en la navegación de robots utilizada en ROS donde cada objeto tiene una posición referencial, normalmente referenciada a la base móvil. En el caso del robot propuesto en la sección 3.2, cada cámara tiene una referencia a su respectivo colector (superior, medio e inferior), y cada colector tiene una referencia al robot. El paquete *TF* nativo en ROS [46], permite realizar un seguimiento de estas coordenadas de cada objeto en relación con otro y también permite cambiar el sistema de referencia de un punto observado o detectado a uno de los objetos del robot, es decir, la ubicación la referencia de un objeto obtenido con la cámara 3D se puede transformar a la referencia de ubicación del colector, así mismo también se puede pasar al robot por transitividad (multiplicación de matrices de traslación y rotación). Finalmente, utilizando el paquete ROS AMCL [47] se puede obtener una estimación de la posición del robot en un entorno utilizando sensores como lidars o cámaras 3D. En el caso del robot, el entorno es el mapa de la tienda, es decir, tiene una referencia al mapa de la tienda en todo momento de su recorrido.

Al final de este bloque se obtendrá la posición de un punto observado por la cámara RGB-UHD en referencia al mapa de la tienda. Esto se utiliza para transformar las ubicaciones en las detecciones de la referencia de imagen RGB-UHD a la referencia del mapa de la tienda, que tendrá un formato [X Y Z], donde X e Y representan el plano 2D del mapa de la tienda y Z es la altura (distancia del suelo a la etiqueta de precio). A esta salida se la denominará Localización de IEPs relativas al mapa de la tienda.

Agrupación de instancias de etiquetas de precio según su localización relativa al mapa de la tienda: Este bloque toma como entrada las localizaciones de IEPs relativas al mapa de la tienda. El enfoque de este bloque es de procesar las localizaciones de las IEPs en referencia al mapa de la tienda para

generar agrupaciones considerando la proximidad de las etiquetas de precio. Se sabe que existe redundancia entre los colectores y entre los pasos, por lo que es muy probable que muchas IEPs estén muy cerca y se agrupen. Esto, es un error pero será validado en bloques posteriores evitando sobrecarga en éste.

Esta agrupación se realiza para un solo lado de un pasillo, dado que este presenta la mejor unidad posible de usar sin que los datos se lleguen a mezclar o segmentar de manera incorrecta generando duplicados innecesarios. El algoritmo utilizado para este agrupamiento es DBSCAN, (implementación del paquete Scikit-learn [48]). Este algoritmo permite agrupar los datos sin necesidad de indicar un número de clúster, lo cual favorece a este proceso ya que no hay forma de saber de antemano cuántas etiquetas de precio reales habrá en cada pasillo. La parametrización del algoritmo se realiza utilizando la metodología propuesta en [49], que explica que el parámetro EPS (distancia máxima entre dos muestras) se puede obtener utilizando los vecinos más cercanos en los datos y encontrando el codo de la función generada. El segundo parámetro es MIN-SAMPLES (número de muestras en una vecindad para que un punto se considere un punto central), el cual se establece en el valor de 1 en este trabajo, dado que un clúster podría ser una sola IEPs.

Para evaluar este algoritmo, las salidas (clústeres) se pasaron por 2 algoritmos de puntuación implementados en el paquete Scikit-learn. Rand Index (RI) [50] y Fixed Rand Index (ARI) [51]. Estos algoritmos evaluan la similitud de los clústeres generados versus los clústeres reales. La puntuación media para el algorítmo RI en los pasillos del conjunto de datos de prueba es 0,9925, mientras que el ARI es 0,7694. ARI usa el orden del clúster para evaluar, mientras que RI no lo hace, por esta razón la métrica RI es mayor que ARI. El uso de estas métricas y que ambas sean relativamente altas dan la noción de que la generación de los clústeres de forma automática es buena. Es necesario indicar que esta primera agrupación es preliminar y puede contener múltiples errores de agrupación dado que solo usar la localización obtenida luego de varias transformaciones no es completamente sólido de usar dado que cada transformación o proyección agrega un delta error. Sin embargo, es lo suficientemente buena la agrupación para tener métricas aceptables. Más adelante otros bloques realizaran un afinamiento en estos clústeres eliminando errores.

La salida de este bloque es un listado de clústeres, que se denominan Clústeres de Etiquetas de Precio Nivel 1 o CEPN 1. Esta salida será utilizada como entrada del subflujo de Selección en conjunto con la salida del subflujo de Imagen-a-Texto. La consolidación de estas salidas se realiza a través del ID de cada IEP.

4.3.3 Subflujo de selección

Este Track realiza el proceso de selección de las mejores IEPs por clúster. Es decir, al final de este subflujo, se eliminarán las redundancias de cada objeto agrupado en un clúster y quedará un único IEP representando al objeto real. Para esto, los clústeres generados en el subflujo de Localización serán filtrados por la confianza obtenida en el reconocimiento de textos de las imágenes IEPs, y luego serán segregadas para evitar la existencia de diferentes productos en un mismo clúster. Finalmente, de cada clúster se elige el mejor IEP utilizando pesos con las confianzas de los items de las imágenes de IEPs, obteniendo una lista de productos que luego se compara con la BDM para encontrar aquellos con precios diferentes y reportarlos.

Filtrado de clústeres: Este es el primer bloque del subflujo de selección. Tiene como entrada los Clúster

Figura 4.6: Ejemplo de densidad de etiquetas de precio en dos bandejas de un pasillo de la tienda del supermercado.

de Etiquetas de Precio Nivel 1 o CEPN 1 (salida del subflujo de Localización) y los textos emparejados de Instancias de Items de etiquetas de precio o Textos Emparejados IIEPs (salida del subflujo de Imagen-a-Texto) consolidadas usando los IDs propios de cada IEP. Este bloque es el encargado de filtrar los IEPs de acuerdo a la confianza de los reconocimientos de los textos de imágenes de IEPs. La elección de los filtros depende de el o los Items del IEP seleccionados para el proceso. En este trabajo, se utilizaron los items Precio y Descripción. En este bloque también cabe aclarar que se pueden utilizar otros items que no hayan sido utilizados en el bloque de Reconocimiento de Textos del subflujo de Imagen-a-Texto. Para este trabajo se mantuvo el mismo esquema y solo se filtraron según los items seleccionados.

En el caso del item Precio, una lectura común sería "$ 1.59" (ver Figura 2.4a). Para que el Precio se lea correctamente, debe tener: Exactamente 2 dígitos decimales y un punto decimal que separe los dígitos enteros de los decimales. El signo de dólar ($) no se considera. Si el texto sigue este formato y la confianza de reconocimiento supera un umbral (en este trabajo, superior a 0,90), el IEP no se elimina.

En el caso del item Descripción, el texto no tiene un formato a seguir, por lo que el filtro seleccionado fue un umbral en su confianza de reconocimiento (en este trabajo, superior a 0,75).

El umbral de confianza de reconocimiento de la Descripción es inferior al del Precio. Esto se debe a la naturaleza del uso de cada tipo de Item de IEP, además de que se requiere tener mayor confianza en el precio ya que se utilizará para comparar al final del proceso y si es incorrecto, generará falsas alertas.

El filtrado ocurre luego de crear los clústeres, de esta manera el algoritmo DBSCAN tiene más datos para obtener automáticamente los argumentos que requiere. Si se hiciera el agrupamiento después del filtrado, habría menos datos, aparte de que la estimación de ubicación con respecto a la tienda del mapa es independiente de la confianza de reconocimiento de cada IEP, es decir, la instancia es real, pero podría haber tenido un ángulo de lectura complicado.

La salida de este bloque son los mismos clústeres, pero cada uno puede tener menos instancias. Esta salida se llama Clúster de Etiquetas de Precio Nivel 2 o CEPN 2.

Segregación de clústeres: Este bloque usa como entrada los Clúster de Etiquetas de Precio Nivel 2 o CEPN 2, que son grupos de IEPs de acuerdo a su proximidad y que también pasaron por un filtro para eliminar los IEPs que no son lo suficientemente confiables en su reconocimiento de texto de imagen de IEP. Agrupándolos por su ubicación, podría haber IEPs de dos o más Etiquetas de precios reales diferentes en el mismo clúster. Esto se dá pues en muchos casos existen etiquetas que están muy cercanas unas de otras en la misma bandeja, en la figura 4.6 se puede ver un ejemplo de este caso. Para solucionar este problema se realiza una etapa de segregación para los CEPN 2. Un ejemplo de esta segregación se puede ver en la figura 4.7, donde un clúster comienza con 8 IEPs agrupados y que previamente ya fueron filtrados. Estos 8 IEPs en realidad son instancias de 3 etiquetas de precio diferentes: $1,99, $0,99 y $1,59 (se indica mediante el precio dado que en este caso particular es el item que más visibilidad tienen las etiquetas mostradas). Para realizar esta segregación, primero se calcula la distancia del coseno entre el primer IEP del clúster y los demás. Las IEPs con una distancia superior a un umbral (0,01 en este trabajo) se separan en otro grupo.

En la figura 4.7, las IEPs que se mantienen en el clúster se muestran en la parte inferior, para este ejemplo el primer IEP seleccionado es el izquierdo superior el cual tiene un valor de $1,99 y que luego de compararlos con los demás se segrega las IEPS que no son similares y ahora se visualizan todas las IEPs de esta única Etiqueta de Precio en la parte inferior, este es clúster inicial pero filtradas todas las IEPs que no eran semejantes. Las IEPs que no son semejantes pasan a un nuevo clúster (en la imagen clúster al lado derecho). Esta segregación no asegura que todos los IEPs separados del siguiente grupo representen el mismo objeto. Por lo tanto, el nuevo clúster se vuelve a segregar indefinidamente hasta que no se pueda crear un nuevo clúster o que todas las IEPs del clúster sean semejantes. En la figura 4.7, se puede ver que el nuevo clúster en el nivel 2 tiene dos productos diferentes (Etiquetas de precio con $1.59 y $0.99). Entonces, la segregación se vuelve a hacer y separa tres IEPs en un nuevo clúster. Después de una tercera segregación, no se crea ningún clúster nuevo, entonces este proceso se detiene.

La salida de este bloque se llama Clúster de Etiquetas de Precio Nivel 3 o CEPN 3, donde todos los clúster tienen IEPs de una sola Etiqueta de Precio real y con valores de confianza de reconocimiento superiores a las cotas definidas.

Selección de mejor instancia de etiqueta de precio por clúster: Este bloque usa como entrada los Clúster de Etiquetas de Precio Nivel 3 o CEPN 3. El proceso en este bloque es de seleccionar el mejor IEP en cada CEPN 3. Para poder realizar esto, se decidió que la mejor IEP sea escogida teniendo en cuenta una confianza ponderada entre las confianzas de los items Precio y Descripción. La ponderación utilizada es de: 0,75 para Precio y 0,25 para Descripción. El precio tiene más peso ya que es más importante para las etapas finales (generación de alertas por comparaciones) y los textos no cambian como la descripción en el bloque de emparejamiento de textos de imágenes de IEPs en el subflujo de Imagen-a-Texto. Para cada clúster que tiene más de una sola IEP, se calcula esta confianza ponderada y se selecciona el mejor sobre los demás. El mejor IEP se denomina Instancia de Etiqueta de Precio Final o IEPF, y se agrega a una lista llamada Lista de Etiquetas de precio final (LEPF).

Comparación de precios: Este bloque tiene como entrada la salida del bloque anterior denominada Lista de Etiquetas de precio final o LEPF. El proceso en este bloque es el de crear una nueva lista comparando las IEPF con los datos de la BDM, y agregar al listado aquellas etiquetas que no tengan los precios iguales a los de la BDM. Esta lista se denomina Lista de Etiquetas de Precio Obsoletas o LEPO. Es decir, la IEPF de un producto con Descripción X y Precio A, se compara con el mismo producto de la BDM con Descripción X (recordar que al momento de emparejar los textos de las imágenes de las IEPs,

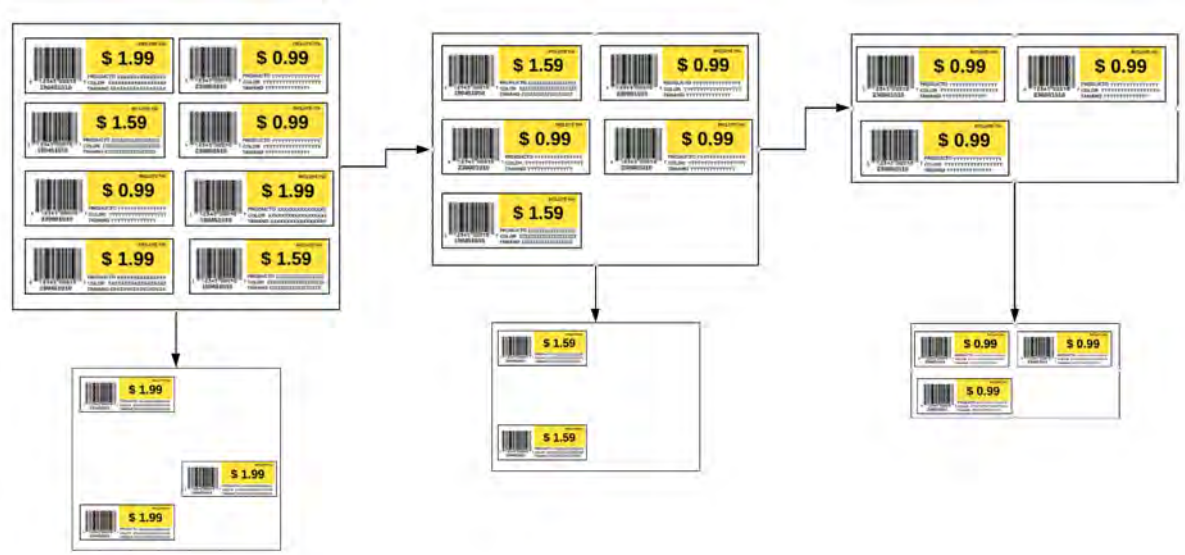

Figura 4.7: Ejemplo de segregación de clústeres. Se observa el resultado de dos procesos de segregaciones. Las etiquetas mantenidas se visualizan en la segunda fila, mientras que las etiquetas segregadas del clúster, son visualizadas a la derecha del clúster inicial.

se realizó una proyección durante el emparejamiento resultando en que toda IEP tenga como descripción siempre un descripción de la BDM) y Precio B. En este caso, si A y B son el mismo texto, la IEP no se guarda en la LEPO, mientras que si A y B son diferentes, si es guardada en la LEPO para luego ser reportada. En el cuadro 4.5 se pueden ver los resultados cuantitativos de la comparación realizada al conjunto de datos de pruebas luego de pasar por todos los subflujos mencionados con anterioridad.

4.3.4 Generación de informes

Este último bloque tiene como entrada la Lista de Etiquetas de Precio Obsoletas o LEPO. La finalidad de este bloque es generar un documento o reporte en cualquier formato definido para ser compartido con el personal operativo necesario. Las alertas que genera son en base a la LEPO.

Tabla 4.5: Resultados finales de los pasillo CERO (izquierda) y UNO (derecha).

		REAL	
		Negativo	Positivo
PREDICIONES Pasillo CERO	Negativo	0	0
	Positivo	1	45

		REAL	
		Negativo	Positivo
PREDICCIONES Pasillo UNO	Negativo	0	6
	Positivo	8	94

4.4 Resultados

Los resultados de este trabajo, se presentan en el cuadro 4.5. Estos son lo suficientemente buenos como para aseverar que el flujo creado para la creación de un reporte de etiquetas de precio obsoletas es efectivo y robusto. También, durante la descripción del detalle de cada bloque del flujo, se indicó donde puede ser mejorado o modificado para ser utilizado con otros diseños de etiquetas de precio de otros supermercados, haciendo que esta solución sea generalizable para cualquier supermercado. El flujo descrito puede ser tomado como un flujo de referencia para la efectiva generación del reporte de etiquetas de precio que los supermercado tanto desean para evitar desperdiciar horas hombre de trabajo.

En el cuadro 4.5 se hace refencia a los pasillos CERO y UNO, junto con los valores cuantitativos de las etiquetas de precio reales y predecidas. El pasillo UNO es más extenso que el pasillo CERO por la totalidad de sus valores, aunque también se podría tener en mente que tiene más concentración de etiquetas de precio por metro lineal de cuerpo de góndola como se vió en la figura 4.6 .

Los errores fueron mínimos y al profundizar en los casos unitariamente, se puede notar que las fallas, si bien son alertas falsas, son ocasionadas por el mismo escenario utilizado en este trabajo. Por ejemplo, el error del pasillo CERO, que es un falso negativo, se produjo debido a un texto de imagen de IEP parcial, específicamente en el item de la Descripción. Este problema es abordado a profundidad en el anexo 5.2. Otro caso particular en el cuadro 4.5 es el caso de los valores que se alertaron pero no existían en el conjunto de datos de prueba. Estos casos ocurren así mismo por emparejamiento errado de descripciones parciales, pero con la singularidad de que se sesgan a productos que no están en la góndola.

Estos resultados entregan un promedio de los pasillos de 94.76% de precisión en la identificación correcta de la etiqueta de precio. Este valor no considera si la etiqueta es obsoleta o no, sino que fue correctamente agrupada con todas sus instancias y que la mejor instancia esté correctamente emparejada con el producto que la EP real indica.

5

Conclusiones

El comercio minorista es un sector importante en el que la IA puede ayudar a reducir las tareas manuales y repetitivas. A la fecha, la IA es lo suficientemente madura como para implementarse en proyectos para resolver problemas reales. En este trabajo se pudo obtener las siguientes conclusiones:

- Se identificaron múltiples actividades manuales y repetitivas en las tiendas de los supermercados ligadas a una macro-tarea denominada Auditaje de Góndolas.
- Se diseñó una propuesta de flujo de trabajo integral utilizando algorítmos de AI y Visión por Computador guiados a automatizar la macro-tarea de Auditaje de Góndolas.
- Se diseñó e implementó un sistema de adquisición de datos útiles para el flujo de trabajo propuesto.
- Se implementó una parte del flujo de trabajo integral propuesto, mostrando la modularidad y robustez del mismo para generar los reportes necesarios para automatizar la macro-tarea de Auditaje de Góndolas.
- Se detalló cada bloque del subflujo de trabajo integral para la identificación de etiquetas de precio de supermercado, dando detalle de los bloques que pueden modificarse o mejorarse según los diseños de las etiquetas aportando a la generalización de la propuesta.
- Se obtuvo una precisión de 94.76% la correcta agrupación e identificación de las mejores etiquetas de precios, facilitando la identificación de las etiquetas de precios obsoletas.

5.1 Lista de contribuciones

Este trabajo ha dado lugar a las siguiente publicaciones en orden cronológico:

- **Moran, E**. and Vintimilla, B. and Realpe, M.. Towards a Robust Solution for the Supermarket Shelf Audit Problem, Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, pages 978-989, VISAPP, ISBN-758-634-7, ISSN 2184-4321, pages 912-919, 2023.
- **Moran, E**. and Vintimilla, B. and Realpe, M.. Towards a Robust Solution for the Supermarket Shelf Audit Problem: Obsolete Price Tags in Shelves. In: Vasconcelos, V., Domingues, I., Paredes, S. (eds) Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. CIARP 2023. Lecture Notes in Computer Science, vol 14469. Springer, Cham. https://doi.org/10.1007/978-3-031-49018-7_19

5.2 Trabajo futuro

Este trabajo presenta una propuesta de flujo de solución para un problema en el sector de la industria minorista. Se ha presentado y validado una de las salidas del flujo propuesto, validando parcialmente la efectividad y factibilidad del flujo completo. Como trabajo futuro, se implementarán los subflujos faltantes como el de detección y reconocimiento de productos, detección de huecos en las góndolas y validación de cumplimiento de planogramas entregando métricas que aporten valor al sector y con reportes que permitan a la fuerza laboral humana de las tiendas (operadores) realizar los arreglos pertinentes de forma rápida y efectiva.

Referencias

[1] T. Bianchi-Aguiar, A. Hübner, M. A. Carravilla, and J. F. Oliveira, "Retail shelf space planning problems: A comprehensive review and classification framework," *European Journal of Operational Research*, vol. 289, no. 1, pp. 1–16, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221720305610

[2] C.-W. Kuo and S.-J. S. Yang, "The role of store brand positioning for appropriating supply chain profit under shelf space allocation," *European Journal of Operational Research*, vol. 231, no. 1, pp. 88–97, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221713004177

[3] G. Kim and I. Moon, "Integrated planning for product selection, shelf-space allocation, and replenishment decision with elasticity and positioning effects," *Journal of Retailing and Consumer Services*, vol. 58, p. 102274, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0969698920312820

[4] X. Drèze, S. J. Hoch, and M. E. Purk, "Shelf management and space elasticity," *Journal of Retailing*, vol. 70, no. 4, pp. 301–326, 1994. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0022435994900027

[5] C. R. M. Ph.D. and M. F. M.A., ""i saw it in the movies": Exploring the link between product placement beliefs and reported usage behavior," *Journal of Current Issues & Research in Advertising*, vol. 24, no. 2, pp. 33–40, 2002. [Online]. Available: https://doi.org/10.1080/10641734.2002.10505133

[6] D. Drexler and M. Souček, "The influence of sweet positioning on shelves on consumer perception," *Food Packaging and Shelf Life*, vol. 10, pp. 34–45, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214289416301090

[7] S. Pettigrew, K. Mizerski, and R. Donovan, "The three "big issues" for older supermarket shoppers," Oct 2005. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/07363760510623894/full/html

[8] B. Li and D. Wang, "Configuration issues of cashier staff in supermarket based on queuing theory," Jan 1970. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-16339-5□44

[9] B. Jedlickova, "Vertical issues arising from conduct between large supermarkets and small suppliers in the grocery market: Law and industry codes of conduct," Apr 2016. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract□id=2764726

[10] F. Cochoy and B. Soutjis, "Back to the future of digital price display: Analyzing patents and other archives to understand contemporary market innovations," *Social Studies of Science*, vol. 50, no. 1, pp. 3–29, 2020, pMID: 31630628. [Online]. Available: https://doi.org/10.1177/0306312719884643

[11] F. Tao, L. Wang, T. Fan, and H. Yu, "Rfid adoption strategy in a retailer-dominant supply chain with competing suppliers," *European Journal of Operational Research*, vol. 302, no. 1,

pp. 117–129, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221721010146

[12] M. Škiljo, P. Šolić, Z. Blažević, and T. Perković, "Analysis of passive rfid applicability in a retail store: What can we expect?" *Sensors*, vol. 20, no. 7, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/7/2038

[13] G. Cilloni, R. Leporati, A. Rizzi, and G. Romagnoli, "State of the art of item-level rfid deployments in fashion and apparel retail," *International Journal of RF Technologies*, vol. 10, pp. 65–88, 2019, 3-4. [Online]. Available: https://doi.org/10.3233/RFT-190174

[14] G. Khalil, R. Doss, and M. Chowdhury, "A novel rfid-based anti-counterfeiting scheme for retail environments," *IEEE Access*, vol. 8, pp. 47 952–47 962, 2020.

[15] M. Ahmed, K. A. Hashmi, A. Pagani, M. Liwicki, D. Stricker, and M. Z. Afzal, "Survey and performance analysis of deep learning based object detection in challenging environments," *Sensors*, vol. 21, no. 15, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/15/5116

[16] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3d object detection methods for autonomous driving applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3782–3795, Oct 2019.

[17] L. B. V. Miguel Realpe, Boris Xavier Vintimilla, "Towards fault tolerant perception for autonomous vehicles: Local fusion," *2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pp. 253–258, 2015.

[18] A. Raghunandan, Mohana, P. Raghav, and H. V. R. Aradhya, "Object detection algorithms for video surveillance applications," in *2018 International Conference on Communication and Signal Processing (ICCSP)*, April 2018, pp. 0563–0568.

[19] T. Hoeser, F. Bachofer, and C. Kuenzer, "Object detection and image segmentation with deep learning on earth observation data: A review—part ii: Applications," *Remote Sensing*, vol. 12, no. 18, 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/18/3053

[20] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *Digital Signal Processing*, vol. 126, p. 103514, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1051200422001312

[21] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. da Silva, "A comparative analysis of object detection metrics with a companion open-source toolkit," *Electronics*, vol. 10, no. 3, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/3/279

[22] X. Zhou, X. Xu, W. Liang, Z. Zeng, S. Shimizu, L. T. Yang, and Q. Jin, "Intelligent small object detection for digital twin in smart manufacturing with industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1377–1386, 2022.

[23] C. Ge, J. Wang, J. Wang, Q. Qi, H. Sun, and J. Liao, "Towards automatic visual inspection: A weakly supervised learning method for industrial applicable object detection," *Computers in Industry*, vol.

121, p. 103232, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361519307559

[24] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2014, cite arxiv:1405.0312Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list. [Online]. Available: http://arxiv.org/abs/1405.0312

[25] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," 2012.

[26] M.-R. Hsieh, Y.-L. Lin, and W. H. Hsu, "Drone-based object counting by spatially regularized regional proposal network," 2017. [Online]. Available: https://arxiv.org/abs/1707.05972

[27] E. Goldman, R. Herzig, A. Eisenschtat, J. Goldberger, and T. Hassner, "Precise detection in densely packed scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[28] M. Ahmed, K. A. Hashmi, A. Pagani, M. Liwicki, D. Stricker, and M. Z. Afzal, "Survey and performance analysis of deep learning based object detection in challenging environments," *Sensors*, vol. 21, no. 15, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/15/5116

[29] F. Chen, H. Zhang, Z. Li, J. Dou, S. Mo, H. Chen, Y. Zhang, U. Ahmed, C. Zhu, and M. Savvides, "Unitail: Detecting, reading, and matching in retail scene," 2022. [Online]. Available: https://arxiv.org/abs/2204.00298

[30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016. [Online]. Available: https://arxiv.org/pdf/1506.01497.pdf

[31] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," 2016. [Online]. Available: https://arxiv.org/abs/1612.08242

[32] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2018.

[33] S. Kant, "Learning gaussian maps for dense object detection," 2020. [Online]. Available: https://arxiv.org/abs/2004.11855

[34] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018. [Online]. Available: https://arxiv.org/abs/1804.02767

[35] X. Pan, Y. Ren, K. Sheng, W. Dong, H. Yuan, X. Guo, C. Ma, and C. Xu, "Dynamic refinement network for oriented and densely packed object detection," 2020. [Online]. Available: https://arxiv.org/abs/2005.09973

[36] G. X. M. L. Jun Yu, Haonian Xie and Q. Ling, "A solution for product detection in densely packed scenes." [Online]. Available: https://trax-geometry.s3.amazonaws.com/cvpr_challenge/detection_challenge_technical_reports/1st_A+Solution+for+Product+Detection+in+Densely+Packed+Scenes.pdf

[37] T. Rong, Y. Zhu, H. Cai, and Y. Xiong, "A solution to product detection in densely packed scenes," 2020. [Online]. Available: https://arxiv.org/abs/2007.11946

[38] F. Chen, H. Zhang, Z. Li, J. Dou, S. Mo, H. Chen, Y. Zhang, U. Ahmed, C. Zhu, and M. Savvides, "Unitail: Detecting, reading, and matching in retail scene," 2022. [Online]. Available: https://arxiv.org/abs/2204.00298

[39] N. Katuk, K. R. Ku-Mahamud, and N. H. Zakaria, "A review of the current trends and future directions of camera barcode reading," *Journal of Theoretical and Applied Information Technology*, vol. 97, no. 8, pp. 1992–8645, 2019.

[40] M. A. Bantahar, S. A. Al-Gailani, and A. A. Salem, "An automatic light control system for camera barcode reader," in *Mahyuddin*, M. N. N. M. and M. S. a. N. R., Eds. Vision, Signal Processing and Power Applications. Lecture Notes in Electrical Engineering, vol 829. Springer, Singapore: Proceedings of the 11th International Conference on Robotics, 2022. [Online]. Available: https://doi.org/10.1007/978-981-16-8129-5□25

[41] R. Brylka, U. Schwanecke, and B. Bierwirth, in *Camera Based Barcode Localization and Decoding in Real-World Applications*. 2020 International Conference on Omni-Layer Intelligent Systems (COINS, 2020.

[42] G. Jocher, "Yolov5: by ultralytics (version 7.0) [computer software]," 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3908559

[43] Google., "Google vision api." [Online]. Available: https://cloud.google.com/vision/docs/apis?hl=es-419

[44] R. Noetic. [Online]. Available: http://wiki.ros.org/noetic

[45] R. A. tag alvar, *An open source AR tag tracking library*. [Online]. Available: http://wiki.ros.org/ar□track□alvar

[46] R. Tf, *Multi-coordinate frame Tracking over time package*. [Online]. Available: http://wiki.ros.org/tf

[47] R. Amcl, *Probabilistic localization System Package*. [Online]. Available: http://wiki.ros.org/amcl

[48] D. Density-Based, *Spatial Clustering of Applications with Noise*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

[49] N. Rahmah and I. Sukaesih, "Sitanggang 2016 iop conf," *Ser.: Earth Environ. Sci.*, vol. 10, no. 1088, pp. 1755–1315.

[50] R. I. Algorithm, *Computes a similarity measure between two clusterings*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.rand□score.html

[51] A. R. I. Algorithm, *Rand index adjusted for chance*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted□rand□score.html

Apéndices

Anexo A

Datos resultantes del proceso del subflujo de selección sobre un cluster. Se debe notar que una instancia tiene el siguiente formato:

```
[   "CLUSTER REAL",Confianza de etiqueta de precio,
        ["Texto de Precio",Confianza de Precio],
        ["Texto de Descripción",Confianza de Descripción],
        ["Texto de Código",Confianza de Código],
],
```

Se inicia con el cluster CEPN 1 y 2, este caso particular no tuvo instancias filtradas porque los reconocimientos tienen confianzas de lectura altas. Tener en cuenta que "000053" es la numeración del cluster que se obtuvo etiquetando los datos manualmente (verdad fundamental o GT).

```
"000029": [ [   "000053",0.8842,
                    ["5.99",0.9596],
                    ["VINO CASTELLO BIANCO 750",0.8500],
                    [null,0]
            ],
            [   "000053",0.9625,
                    ["5.99",0.9745],
                    ["VINO CASTELLO BIANCO 750 ML ROSADO",0.9228],
                    ["249523000",0.9932]
            ],
            [   "000053",0.9087,
                    ["5.99",0.9818],
                    ["VINO CASTELLO BIANCO 750",0.8858],
                    [null,0]
            ],
            [   "000053",0.9632,
                    ["5.99",0.9727],
                    ["VINO CASTELLO BIANCO 750 ML ROSADO",0.9307],
                    ["249523000",0.9933]
            ],
            [   "000053",0.8981,
                    ["5.99",0.9650],
                    ["VINO CASTELLO BIANCO 750",0.8904],
                    [null,0]
            ],
            [   "000053",0.9649,
                    ["5.99",0.9666],
                    ["VINO CASTELLO BIANCO 750 ML ROSADO",0.9538],
                    ["249523000",0.9930]
            ]]
```

A continuación, el CEPN 3 donde ya se puede notar que el cluster enumerado como "000029" se segregó a otro cluster denominado "000068". En el CEPN 3 también se aprecia la selección del mejor ("best") de cada cluster.

```
"000029": {
            "best": [
                        "000053",0.9087,
                            ["5.99",0.9818],
                            ["VINO CASTELLO BIANCO 750 ML BLANCO",0.8858],
                            [null,0]
                    ],
            "others": [
                        [   "000053",0.8842,
                                ["5.99",0.9596],
                                ["VINO CASTELLO BIANCO 750 ML BLANCO",0.8500],
                                [null,0]
                        ],
                        [   "000053",0.8981,
                                ["5.99",0.9650],
                                ["VINO CASTELLO BIANCO 750 ML BLANCO",0.8904],
                                [null,0]
                        ]
                      ]
          },
...
"000068": {
            "best": [
                        "000053",0.9649,
                            ["5.99",0.9669],
                            ["VINO CASTELLO BIANCO 750 ML ROSADO",0.9538],
                            ["249523000",0.9930]
                    ],
            "others": [
                        [   "000053",0.9625,
                                ["5.99",0.9748],
                                ["VINO CASTELLO BIANCO 750 ML ROSADO"0.9228],
                                ["249523000",0.9932]
                        ],
                        [   "000053",0.9632,
                                ["5.99",0.9727],
                                ["VINO CASTELLO BIANCO 750 ML ROSADO",0.9307],
                                ["249523000",0.9933]
                        ]
                        ]
          },
```


Este caso presentado es el único error del conjunto de datos de pruebas del pasillo CERO, el cual corresponde al cluster denominado 000053. En la figura 1, se pueden observar todas las IEPs del cluster 000053. Este cluster contiene IEPs provenientes de dos colectores y 3 pasos del robot. Se puede notar rápidamente que todas las IEPs pueden ser leídas correctamente por los resultados mostrados en el CEPN 1. Lamentablemente, en los IEPs 1a, 1c y 1e, una parte importante de la descripción no puede ser

(a) IEP 1 (b) IEP 2 (c) IEP 3

(d) IEP 4 (e) IEP 5 (f) IEP 6

Figura 1: Imágenes de etiquetas de precios de un cluster del conjunto de datos de pruebas. Cluster No. 000053

observada. Esto genera el error que se observa en el CEPN 3, donde por segregación se separan estos IEPs en dos cluster (000029 y 000068). Se puede notar que todos los IEPs que tienen este problema se mantuvieron agrupados en un solo cluster. Esto facilmente da a entender que por el emparejamiento realizado en una descripción leída aún de manera correcta, puede darse un desviamiento hacia un producto similar. En este caso en vez de guiarse al producto con descripción "VINO CASTELLO BIANCO 750 ML ROSADO", los IEPs 1a, 1c y 1e se sesgaron hacia "VINO CASTELLO BIANCO 750 ML BLANCO". El texto visible en las imágenes de las IEPs 1a, 1c y 1e es "VINO CASTELLO BIANCO 750", lo cual es correcto por ser lo único visible en las etiquetas de precio. Esto sucede por el diseño de la etiqueta de precio y casos en donde se descubrió que la etiqueta se imprimió mal, haciendo que el diseño este ligeramente más abajo de lo normal; esto provoca que al momento de tomarle la foto, por el tipo de sostén que tiene la bandeja en su frente, parte de la descripción quede ocluída.