

Detección computacional de objetos para cumplimiento de planogramas en OXXO

Autores

-Barredez Ríos, Carlos Andrés
-Sánchez Pérez, Raúl Andrés

Profesor: Balderas Silva, David Christopher

-Delgado Ríos, Leonardo
-Hure, Marin Joseph

Profesor: López Bernal, Diego

Introducción

La problemática principal radica en el tiempo y esfuerzo significativo que los colaboradores en tienda dedican al **proceso manual denominado "cumplimiento de planograma"**. Este proceso implica verificar el acomodo adecuado de los productos en los anaqueles.

Este proyecto utiliza **visión computacional para agilizar y mejorar el cumplimiento de planograma en tiendas**. Reduciendo la carga manual, la solución detecta y notifica desviaciones y proporciona instrucciones precisas, optimizando así la eficiencia y el cumplimiento operativo

Recolección de información

La recolección de imágenes desde recursos presenciales a recursos digitales fue un **aspecto crucial en nuestro proyecto**, ya que se necesitaban datos de calidad que al final serían la representación de la eficiencia de nuestra solución

- Se recolectaron aproximadamente **2000 imágenes de 36 productos en total**.
- Se estima una cantidad **de 50-100 imágenes por producto**

Además de lo mencionado se realizó **Data Augmentation** llevando al dataset a contener una cantidad de **más de 3500 imágenes**

Etiquetado de imágenes

Se utilizaron dos herramientas diferentes para etiquetar la gran cantidad de imágenes de nuestro dataset.

Supervisely

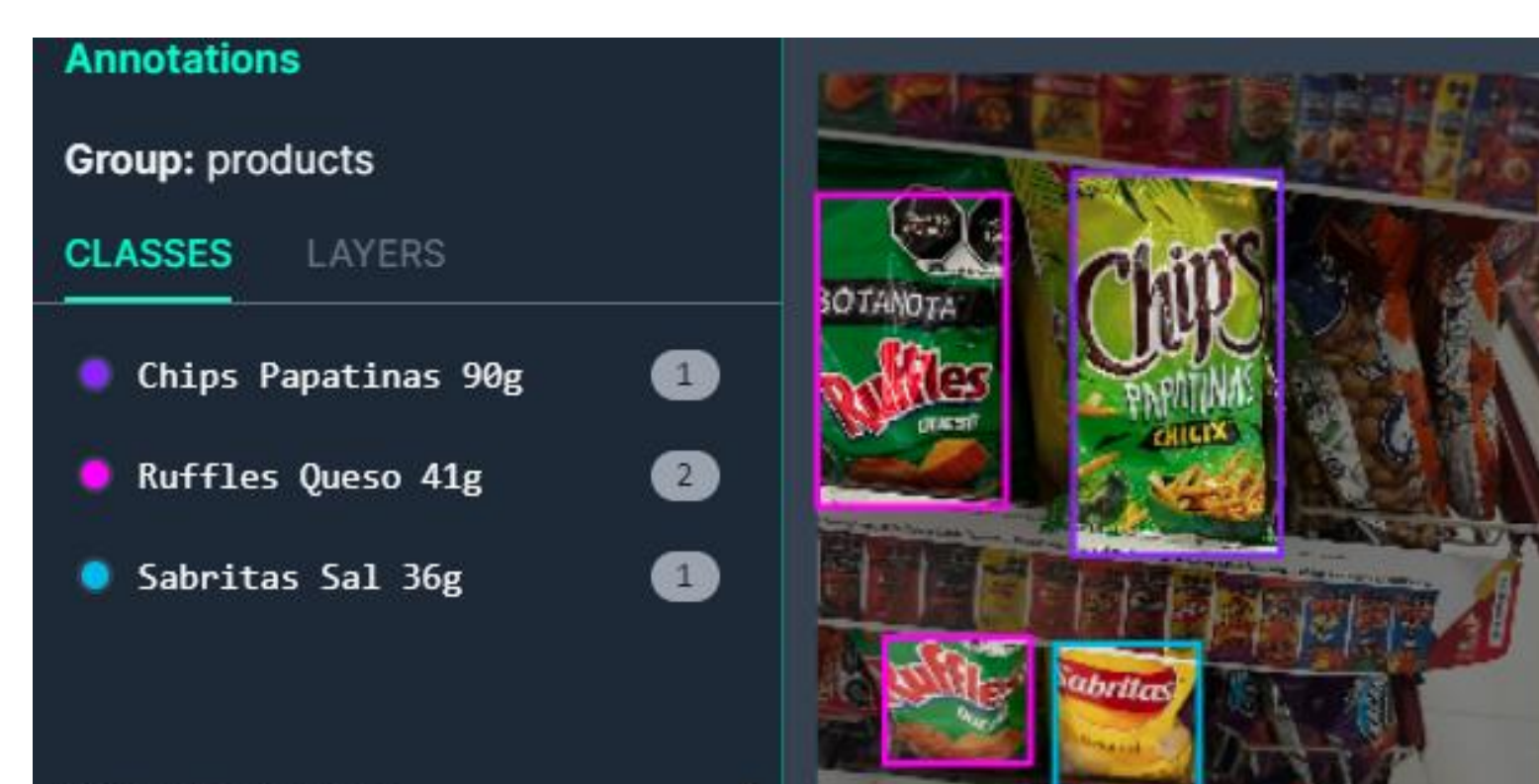
Esta herramienta nos permitía etiquetar los objetos con **mascaras**, llevando a una selección muy precisa



Etiquetado con masks

Roboflow

Esta herramienta utiliza **bounding-boxes** para etiquetar objetos, finalmente su compatibilidad resultó ser la ventaja decisiva



Etiquetado con bounding boxes

TRAIN SET

98%

3254 Images

VALID SET

5%

169 Images

TEST SET

5%

175 Images

Dataset split

Modelo de visión

Se hizo una preselección de tres modelos a desarrollar, con el fin de comparar sus resultados, limitantes y aplicación en términos del proyecto.

Se utilizaron datos de la seccion de papas para esta etapa



OCI Vision nos presentó buenos resultados, sin embargo, **sus limitantes en cuanto a recursos para cuentas gratuitas nos hicieron descartarlo**



Detección OCI Vision



Nuestro modelo CNN utilizo masks en lugar de bounding boxes, sin embargo, su aplicación **se descartó debido a un modelo aún más eficiente**

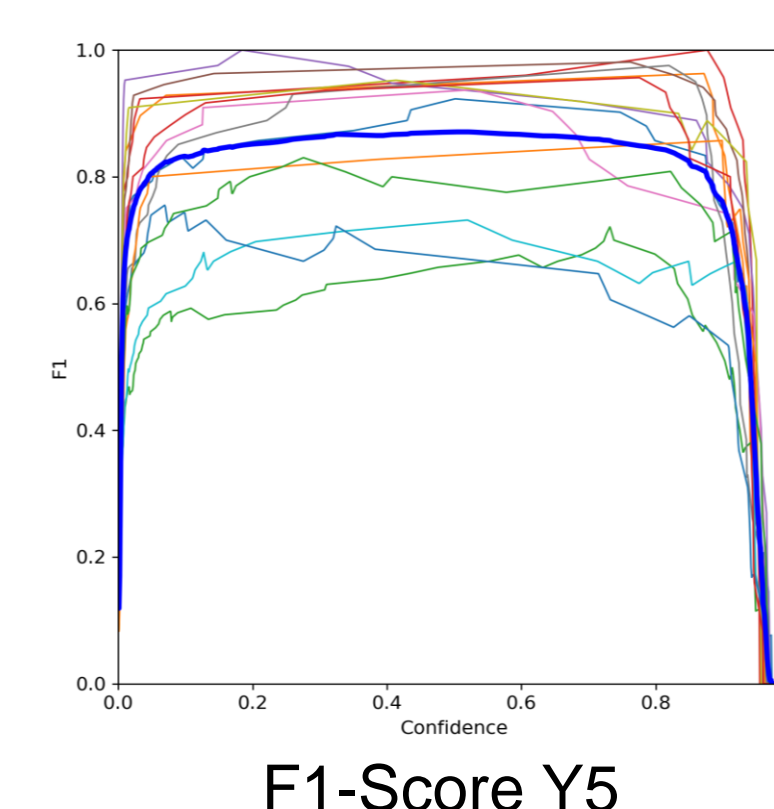


Detección CNN con masks



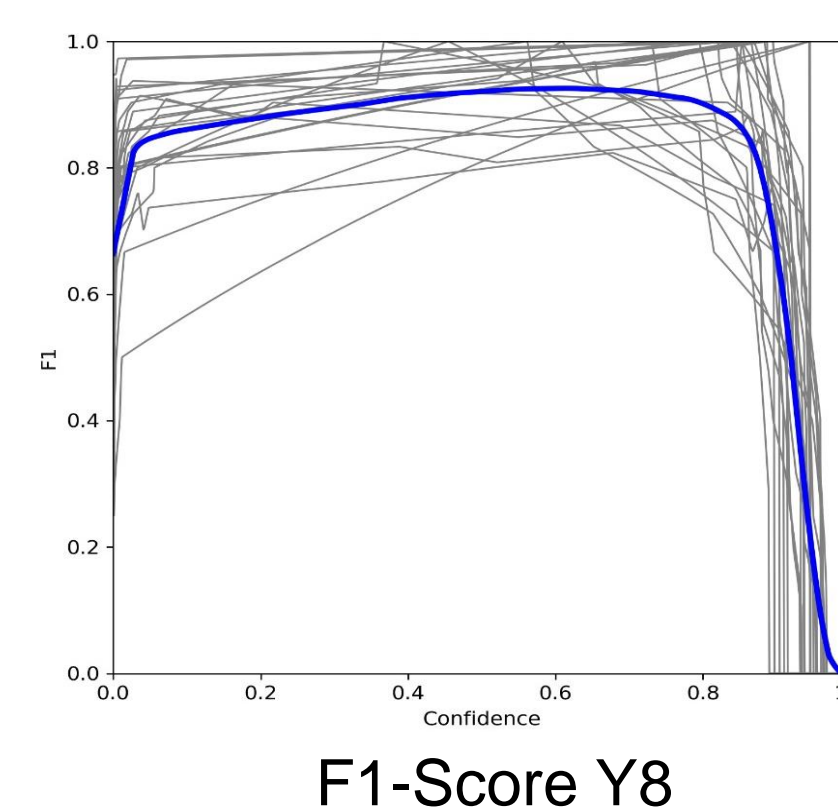
Nuestro modelo seleccionado fue Yolo8, demostrando la mejor capacidad en detección de objetos, igualmente se realizaron pruebas con Yolo5

Yolo5~ 87%

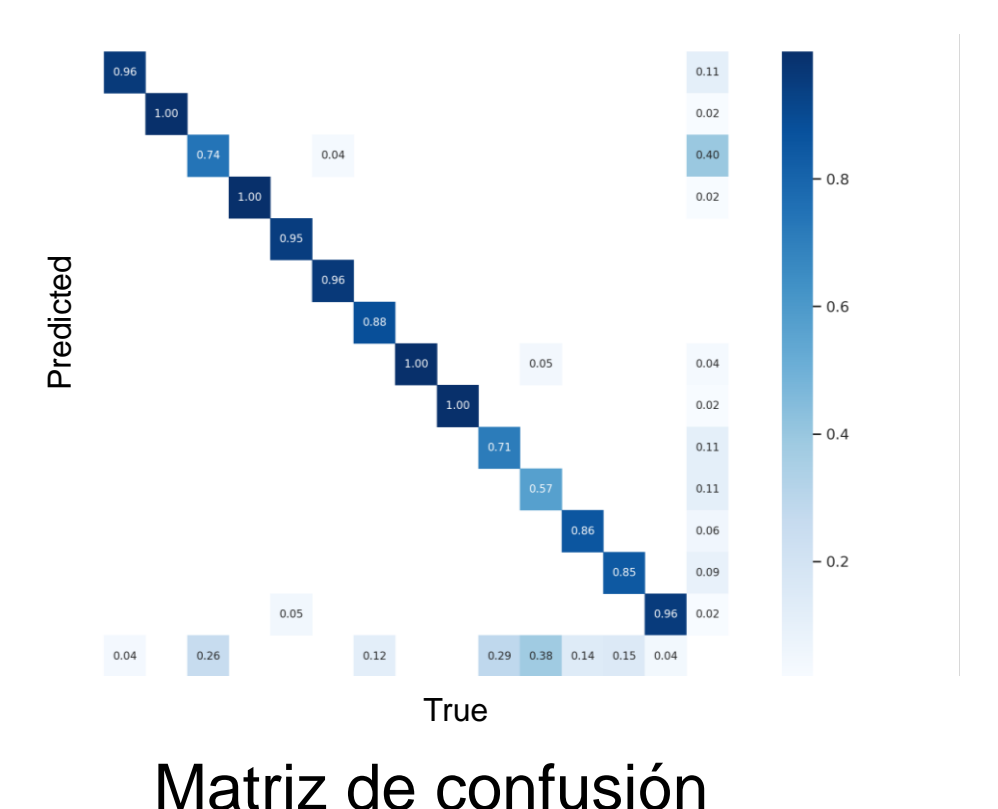


F1-Score Y5

Yolo8~ 93%



F1-Score Y8

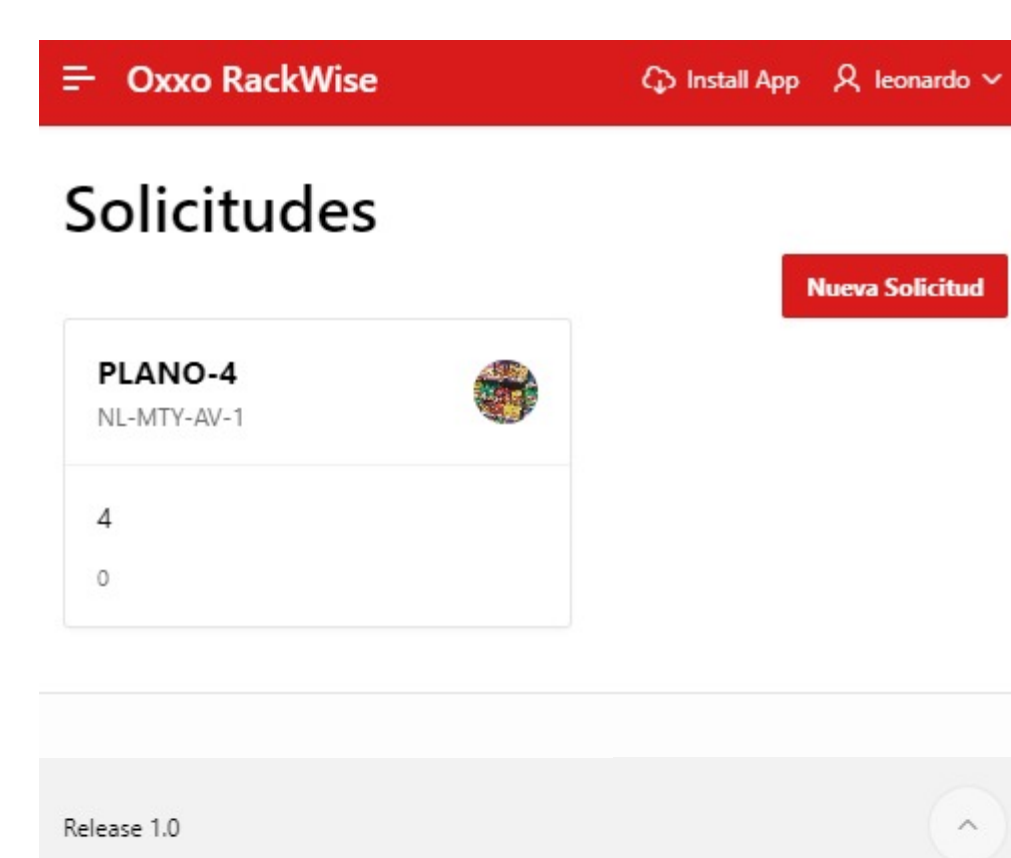


Matriz de confusión

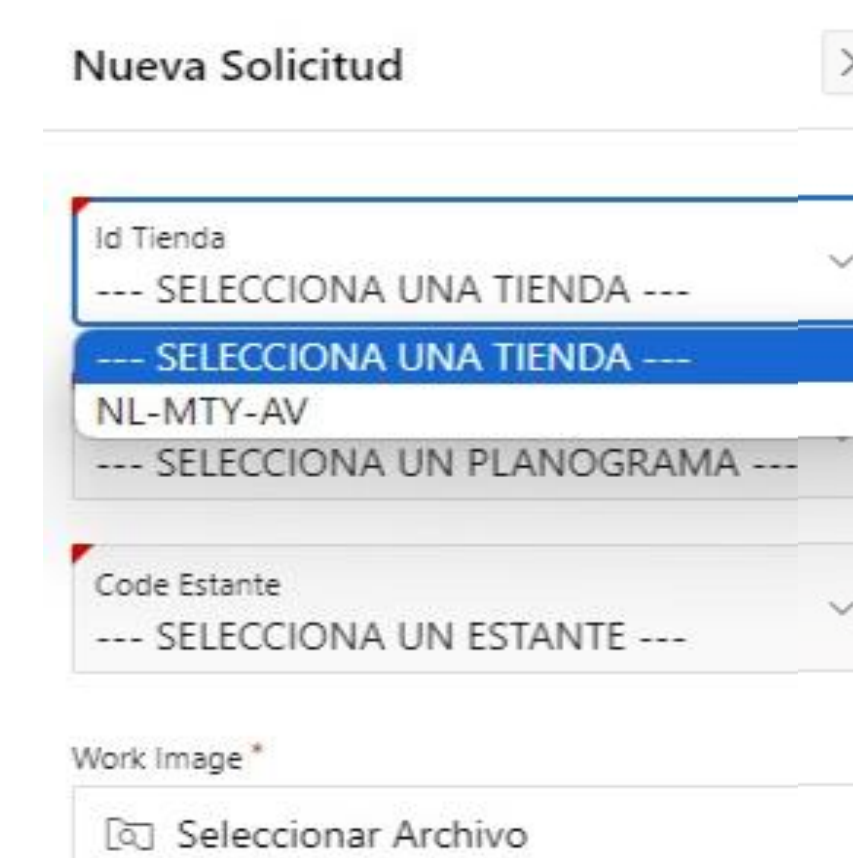
Aplicación

Nuestra aplicación se desarrolló con **el entorno Apex de Oracle** ya que presentaba compatibilidad con nuestro proyecto.

La aplicación es funcional en **pc o dispositivo móvil utiliza una conexión a internet** para registrar solicitudes y posteriormente acceder a los resultados procesados por el modelo computacional donde se realizan sugerencias de acomodo



Board de Solicitudes



Nueva Solicitud de revisión de acomodo



Board de resultados

Revision Resultados

Id MI Work	Result Image	Product Code	Product Count	Result Suggestion	Result Precision
PLANO-4	[datatype]	40	2	Producto detectado no esta en configuracion.	91.5
PLANO-4	[datatype]	10	2	Producto detectado no esta en configuracion.	83.14
PLANO-4	[datatype]	38	5	Producto detectado no esta en configuracion.	87.09
PLANO-4	[datatype]	11	2	Producto detectado no esta en configuracion.	87.86

Revisión de resultados

Conclusiones

Navegando a través de los recursos limitados y mediante la clasificación de prioridades se logró entregar una solución inicial para el problema, las áreas de mejora son abundantes, especialmente hablando de cuestiones de arreglo de bugs, mejora de visuales, etc.