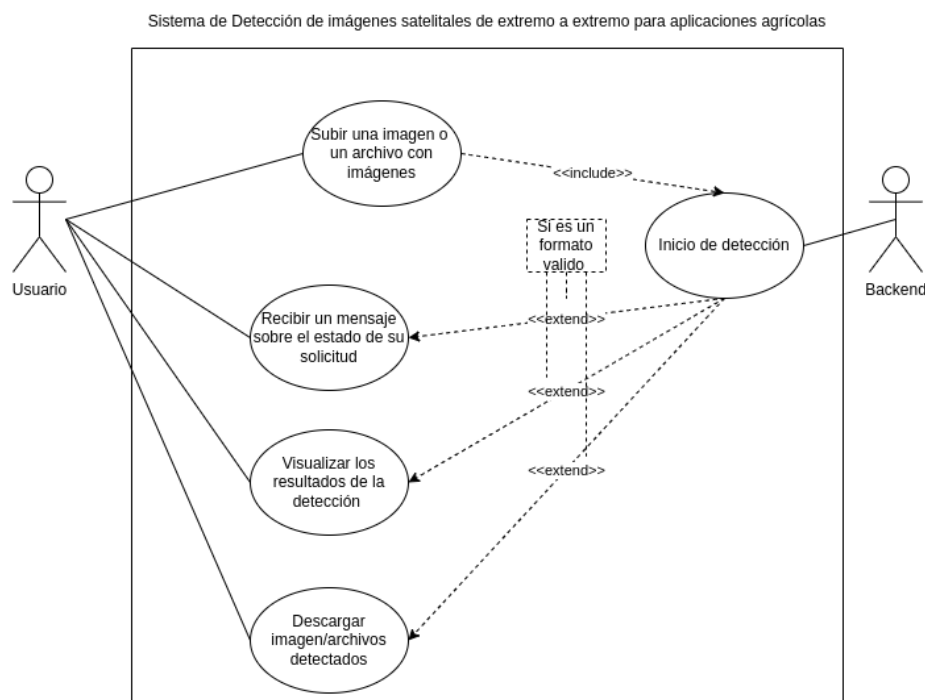


Resumen Proyecto:

“Detector de Pivotes de Riego y Silobolsas en Imágenes Satelitales para aplicaciones agrícolas”

El proyecto es concretamente una página web donde un usuario puede cargar una o varias imágenes satelitales y recibe en poco tiempo un análisis de la misma, en donde se detecta o no la presencia de Pivotes de Riego y/o Silobolsas.

Técnicamente, y a muy grandes rasgos, se trata de un modelo de Machine Learning, el cual corre en un servidor y publica un endpoint, donde la página web realiza las consultas que el usuario solicita.



Tras varias entrevistas con las partes interesadas, los mismos, como conocedores del área de las imágenes satelitales y del sector agropecuario, permitieron:

- Entender las primeras especificaciones para un prototipo.
- Comprender mejor los problemas del sistema de riego que se usa en el campo, como así también el almacenamiento de los cereales, legumbres, entre otros.
- Hacer uso de éste prototipo y sus beneficios como parte de la solución de éstos problemas en un futuro.

De la charla, se plantearon los siguientes problemas:

- Deforestaciones ilegales.
- Asentamientos ilegales y expropiaciones.
- Evasión impositiva en el rubro agropecuario.
- Abuso de los recursos hídricos

La solución propuesta que surgió de la entrevista fue:

- Debido a la fuerte influencia de algoritmos de Machine Learning de detección de objetos, se propuso hacer uso de ésta tecnología para desarrollar una aplicación que provea lo siguiente:
 - Detectar múltiples objetos, es este caso, las silobolsas y los pivotes de riego.
 - Dar la posición X e Y del objeto en la imagen (o su centro) y dibujar un rectángulo a su alrededor.
 - La posibilidad de detectar “a tiempo”. Ésta es una característica que se debe tener en cuenta si por ejemplo se quiere hacer detección en tiempo real sobre vídeo. Esto es posible gracias a la velocidad y precisión para la inferencia que los detectores de objetos proveen.

Para realizar lo anteriormente descrito, se dispone de un conjunto de imágenes satelitales de público acceso que describen diferentes áreas del territorio argentino. Con ellas se plantea hacer uso de la Ciencia de Datos, el análisis estadístico y Machine Learning para implementar un software que permita analizar el contenido de una imagen satelital y devuelva resultados útiles para la búsqueda de infractores, incorporando de esta forma tecnologías innovadoras y altamente populares en la industria al ámbito público.

Ante las preocupaciones anteriormente planteadas, se propone una solución para satisfacer esas necesidades que no estaban cubiertas, por lo cual se enumeran a continuación las motivaciones que se presentaron para llevarlo a cabo:

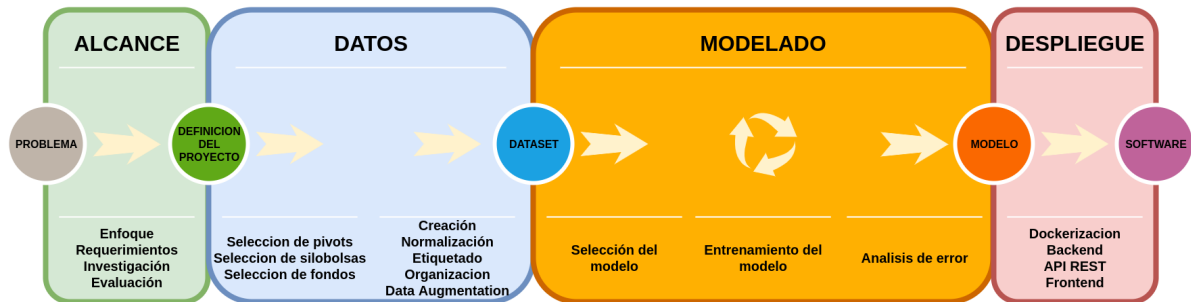
1. Elevar la inteligencia de la toma de acciones automáticas, por ejemplo, en un sistema de Dirección General de Rentas.
2. Incorporar tecnologías novedosas y altamente populares en la industria al ámbito agropecuario público.
3. Ayudar a combatir irregularidades o deforestaciones detectándolas en tiempo real. En éste caso, detectando pivotes de riego y silobolsas, ambos utilizados ampliamente en el rubro agrícola.
4. Adquirir conocimientos sobre Deep Learning e Inteligencia Artificial y aplicar dichos conocimientos a desarrollar un prototipo para la detección de imágenes satelitales en un periodo de tiempo corto.

En cuanto al cómo se implementó este proyecto, se pueden nombrar algunos puntos claves:

- El modelo de Machine Learning se llama YOLO (YouOnlyLookOnce), y es un modelo de Computer Vision para detección de objetos re-entrenable muy performante, es decir, realiza la detección en un corto periodo de tiempo, o como suele decirse, “en tiempo real” y que al día de la fecha es el estado del arte en object detection.
- El proyecto se desarrolló e implementó en su totalidad en la supercomputadora de la facultad Nabucodonosor (o Matrix), en particular para hacer uso de su gran poder de cómputo para entrenar el modelo para nuestro dataset.

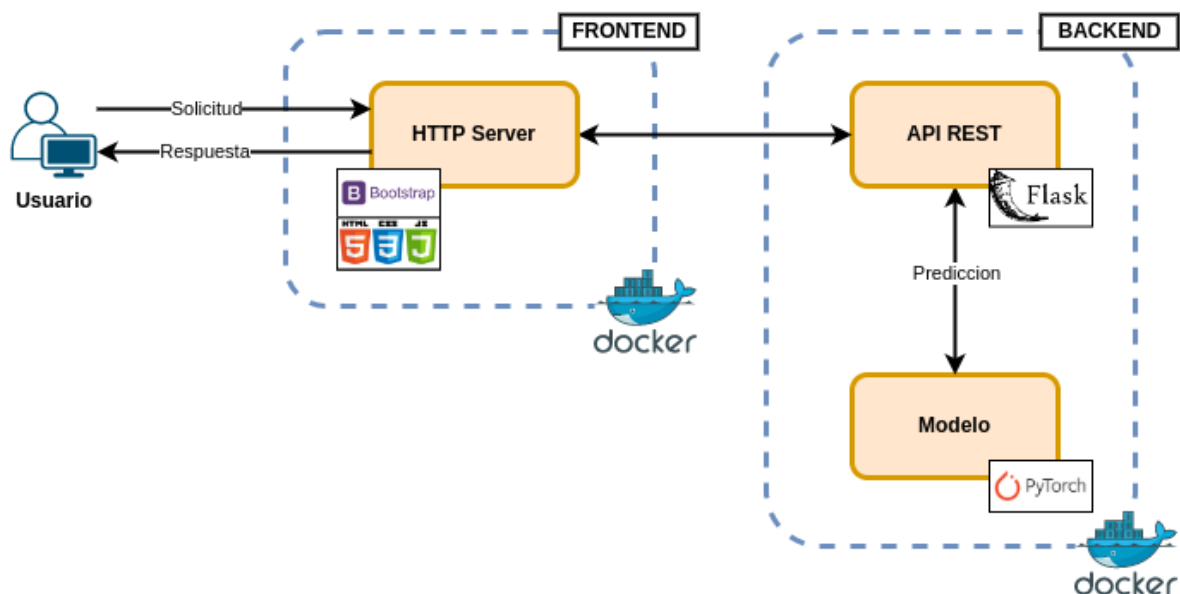
- El software se dockerizó para ser fácilmente desplegable o exportable en otros sistemas.

El trabajo está estructurado según el siguiente Workflow:

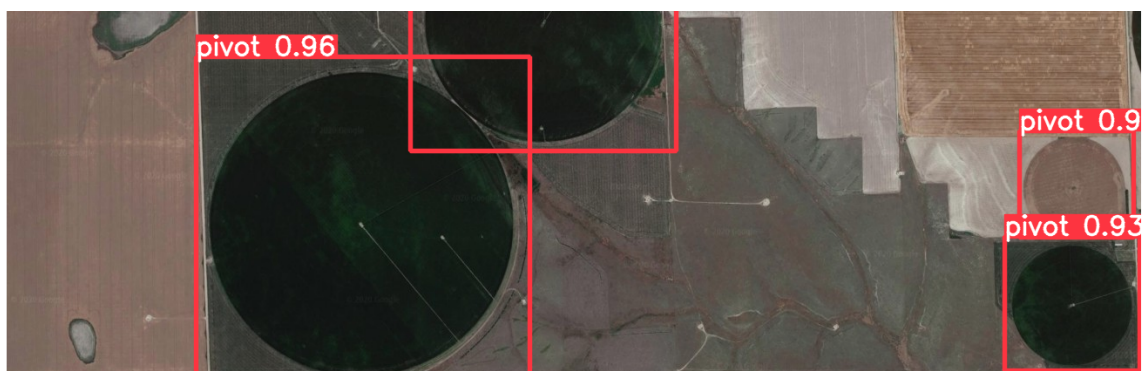


1. **Alcance:** Primero se definió el alcance de nuestro proyecto, para quien estaba enfocado y todo lo necesario para llevarlo a cabo. Se decidió comenzar por una investigación y posterior evaluación de los algoritmos más utilizados para la detección de objetos en tiempo real, con el afán de lograr tiempos de detección reducidos.
2. **Datos:** Se continuó con la elección de los datos a utilizar, en este caso imágenes de riegos por pivot y silobolsas con vista satelital. Se siguió con la creación, normalización y etiquetado de un conjunto de datos, para formar el dataset adecuado.
3. **Modelo:** Luego del análisis entre los diferentes modelos más usados, se optó por una primera elección de YOLOv3, y se procedió con su entrenamiento con los datos previamente seleccionados. Debido a que no se obtuvieron los resultados esperados, se decidió emplear el algoritmo YOLOv5, que además de tener varias mejoras respecto a la versión 3 resultó ser más efectivo para nuestro problema y se obtuvo un mejor rendimiento.
4. **Despliegue:** Después se procedió a desarrollar un prototipo de la aplicación con el fin de evaluar la implementación y utilidad de la propuesta. Para eso, el desarrollo se dividió en tres partes:
 - a. La primera parte consistió en la configuración del Algoritmo YOLOv5 en un entorno virtual contenido en un Docker con los requerimientos y librerías necesarias. Se configuraron 2 entornos virtuales: uno para realizar los entrenamientos del algoritmo y dejar corriendo el contenedor Docker con el backend (modelo YOLOv5 + API) y otro entorno con el frontend (página web).
 - b. La segunda parte consistió en la programación de la API con la ayuda del Framework Flask para el backend. Se implementaron las siguientes funcionalidades:
 - i. Subir una imagen nueva o un archivo comprimido con un conjunto de imágenes para ser analizadas.
 - ii. Consultar la información resultante de cada imagen subida.
 - iii. Consultar la información y los parámetros del modelo utilizado.

- iv. Otorgar la opción de descargar la/s imagen/es analizada/s junto con los resultados de la predicción.
- c. En la última parte, se desarrolló una aplicación web (Frontend) para que el usuario pueda interactuar con el sistema a través de una interfaz gráfica. Con los siguientes lenguajes de programación: JavaScript, HTML y CSS y el framework Bootstrap, se le dio estructura y estilo a la página, dando una simple retroalimentación visual al usuario para indicar los resultados de la detección y del análisis.



A continuación se presenta una imagen del software tras realizar una inferencia sobre la imagen cargada por el usuario a través de la página web:



[Download](#)

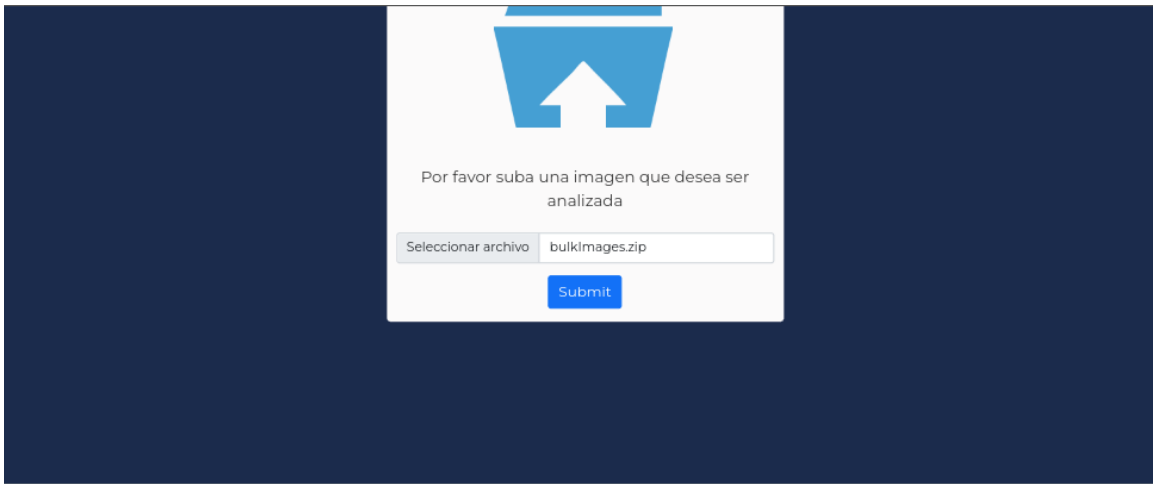
Información del Análisis

| Nombre | Tiempo Total | Tiempo de Inferencia | Tamaño de la imagen | Pivotes Detectados | Silobolsas Detectadas |
|------------|--------------|----------------------|---------------------|--------------------|-----------------------|
| riego1.png | 60.7ms | 0.061s | 416x640 px | 5 | 0 |

El usuario puede visualizar la imagen ya analizada y el resultado de la detección, remarcando los cuadros delimitadores, encerrando a los objetos detectados e indicando su tipo.

Además, se detalla el historial y los resultados de la detección. La figura anterior muestra el resultado en caso de haber subido una sola imagen.

En la siguiente figura, se muestra el ejemplo de subir un archivo comprimido con varias imágenes y el resultado de la detección:



Resultados del Análisis

Información del Modelo de Machine Learning

| Modelo | Tamaño de Imagen de Entrada | Umbral de Confianza | Umbral de IoU |
|--------------|-----------------------------|---------------------|---------------|
| YOLOv5 Small | 640x640 px | 0.65 | 0.45 |

Download

Información del Análisis

| Nombre | Tiempo Total | Tiempo de Inferencia | Tamaño de la Imagen | Pivotes Detectados | Silobolsas Detectados |
|-----------------|--------------|----------------------|---------------------|--------------------|-----------------------|
| bolsa1.png | 54.2ms | 0.053s | 320x640 px | 0 | 0 |
| pivot_silo1.png | 54.2ms | 0.053s | 352x640 px | 1 | 0 |
| pivot_silo2.png | 54.2ms | 0.054s | 384x640 px | 0 | 0 |
| pivot_silo3.png | 54.2ms | 0.052s | 352x640 px | 0 | 0 |
| riego1.png | 54.2ms | 0.054s | 416x640 px | 5 | 0 |
| riego2.png | 54.2ms | 0.060s | 512x640 px | 3 | 0 |