Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Querétaro

Momento de Retroalimentación

Reto Datos

Autores:

A01368818 Joel Sánchez Olvera

A01661090 Juan Pablo Cabrera Quiroga

A01704076 Adrián Galván Díaz

A01708634 Carlos Eduardo Velasco Elenes

A01709522 Arturo Cristián Díaz López

TC3007C.501

Inteligencia artificial avanzada para la ciencia de datos II

Fecha:

14 de Octubre del 2024

Introducción

Este documento aborda el desarrollo de un proyecto enfocado en la implementación de inteligencia artificial avanzada para la detección y clasificación de posiciones de vacas en imágenes, un desafío planteado por CAETEC. En el contexto de la ciencia de datos, este tipo de proyectos demanda tanto el procesamiento de grandes volúmenes de información como el uso de modelos de aprendizaje profundo que puedan optimizar tareas repetitivas y mejorar la precisión en el análisis de imágenes. Para lograr estos objetivos, se emplean frameworks de modelos para machine learning, así como técnicas de limpieza y preparación de datos que aseguran la calidad y relevancia del conjunto de datos procesado.

El trabajo se divide en diversas etapas, cada una de las cuales aborda una parte fundamental en el desarrollo de un sistema de machine learning. Se inicia con una descripción de las herramientas y tecnologías empleadas, justificando su uso y relevancia. Posteriormente, se presenta el modelo de almacenamiento y procesamiento de datos utilizado, junto con una propuesta de escalabilidad en el futuro. La tercera sección se centra en la extracción, limpieza y carga de los datos, destacando la importancia de la organización y preparación de las imágenes antes de su análisis. Adicionalmente, se aplica un esquema de validación k-fold cross-validation para evaluar el desempeño del modelo, evitando problemas de sobreajuste y garantizando una mayor robustez en la predicción. Por último, se analiza la viabilidad de emplear un enfoque orientado a Big Data en caso de que el proyecto escale, considerando el potencial de crecimiento en el número de imágenes y otros datos asociados.

Este proyecto representa una oportunidad para aplicar técnicas avanzadas de inteligencia artificial en el análisis de imágenes y muestra cómo estas tecnologías pueden optimizar tareas en industrias que requieren un procesamiento eficiente de datos visuales. A lo largo del documento, se examinan cada uno de estos aspectos, proporcionando una visión detallada del proceso de implementación y justificación de cada decisión técnica.

1. **Herramientas y tecnologías que se usarán para trabajar con los datos (Justificación)**

Fase	Objetivo	Herramientas	Justificación
Extracción	Obtener todos los datos posibles garantizando la integridad y calidad de los datos extraídos para satisfacer los requerimientos iniciales del proyecto.	One Drive	Usamos OneDrive principalmente debido a que en esa plataforma se encontraba el dataset original del socio.
Transformación	Aplicar procesos de limpieza, normalización y de datos para estandarizar los formatos y asegurar la coherencia, haciendo que los datos sean estructurados y consistentes para el análisis o modelado posterior.	Python Roboflow	**Bounding Box** Para el bounding box, se utilizó la herramienta de Roboflow ya que ésta provee una interfaz gráfica para hacer el etiquetado de las imágenes de manera más eficiente. Esto nos ayudó para crear las bounding boxes de las vacas en cada una de las imágenes del dataset original, lo cual hizo el proceso mucho más eficiente y esa fue la razón principal por la cual decidimos utilizar Roboflow. **Clasificador** Para el modelo clasificador modificamos el tamaño de la imagen a 224 x 224 para una normalización estándar de las mismas. Así mismo se estandarizó el brillo de la imagen convirtiendo los valores de los píxeles a un tipo de dato float32 y después dividiéndolos entre 255 para que queden en un rango entre 0 y 1. Esto es común en el preprocesamiento de imágenes, ya que ayuda a mejorar el

			rendimiento y la estabilidad de muchos modelos de machine learning.
Load (Carga)	Almacenar los datos transformados en un sistema de destino de manera eficiente y segura, asegurando su disponibilidad para consultas y análisis, y minimizando la latencia en el acceso a la información final.	Drive	
Modelado		Tensor Flow Pytorch	**Bounding Box** **Clasificador**
Análisis de patrones en la arena	Determinar la existencia de patrones repetitivos o estructuras organizadas en la superficie de las camas de arena que pudieran afectar el comportamiento de descanso del ganado	Python (OpenCV, NumPy, SciPy, Matplotlib, Seaborn)	Para el análisis de patrones se implementaron múltiples técnicas de procesamiento de imágenes y análisis de textura: 1) Análisis de textura mediante GLCM para evaluar características como contraste, homogeneidad y correlación. 2) Detección de bordes usando métodos Sobel, Canny y Laplaciano para identificar estructuras. 3) Análisis de periodicidad y escala para detectar patrones repetitivos. 4) Mapeo de direcciones de gradiente para estudiar orientaciones preferentes. Esta combinación de técnicas proporcionó un análisis comprehensivo que permitió evaluar la superficie desde múltiples perspectivas, asegurando la detección de cualquier patrón significativo que pudiera afectar al ganado.

En el proyecto de CAETEC se utilizarán las siguientes herramientas y tecnologías:

- **Python**: Se utilizará como el lenguaje principal para el procesamiento de datos y entrenamiento de los modelos de Machine Learning.
- **Pandas y Numpy**: Estas bibliotecas serán utilizadas para la manipulación y análisis de los datos.
- **PyTorch y TensorFlow**: Se emplearán para implementar y entrenar modelos de Deep Learning enfocados en la detección de vacas y clasificación de sus posiciones.

La justificación del uso de estas herramientas se basa en su capacidad para manejar eficientemente grandes volúmenes de datos, la facilidad para integrar técnicas avanzadas de machine learning, y la flexibilidad que ofrecen para optimizar tanto la fase de procesamiento de datos como la fase de modelado.

2. Modelo de almacenamiento de los datos

El proyecto utiliza actualmente Google Drive como repositorio de datos para las 9634 imágenes que componen el conjunto de datos. Google Drive proporciona una solución de almacenamiento en la nube que permite un acceso sencillo y rápido a los datos desde cualquier ubicación, lo que es esencial para un equipo colaborativo como el de este proyecto. Las imágenes, que tienen un tamaño promedio de entre 600 y 650 KB, requieren de un almacenamiento que permita tanto la organización como la accesibilidad constante para el análisis y procesamiento.

Sin embargo, para garantizar una mayor escalabilidad y manejo de datos en futuros escenarios, se podría migrar hacia un sistema de almacenamiento más robusto como **Amazon S3 (Simple Storage Service)**. Esta tecnología proporciona un almacenamiento en la nube escalable y seguro, ideal para manejar grandes volúmenes de datos como imágenes o videos. Amazon S3 es utilizado ampliamente por su flexibilidad, alta disponibilidad y durabilidad, lo que asegura que los datos estarán accesibles en cualquier momento y desde cualquier parte del mundo.

<u>Pasos para implementar S3 en el proyecto</u>

1. Crear un Bucket en S3

- Ir a la sección de S3 y crear un nuevo bucket donde se almacenarán las imágenes del proyecto.

- Definir los permisos de acceso para controlar quién puede subir y acceder a las imágenes, asegurando que solo los miembros del equipo autorizado tengan acceso.

2. Cargar los datos:

- Subir las 9634 imágenes existentes al bucket de S3. Esto se puede hacer manualmente a través de la consola de AWS o programáticamente utilizando herramientas como la AWS CLI (Command Line Interface) o el SDK de Python, Boto3.
- Para el futuro, se podría automatizar la carga de nuevas imágenes utilizando scripts en Python que carguen automáticamente cualquier nuevo archivo de imagen al bucket.

3. Configurar permisos y políticas

- Definir políticas de bucket para establecer restricciones de acceso. Por ejemplo, el acceso a los datos podría estar restringido por IP o por los roles de los usuarios.
- Habilitar el cifrado automático de los datos para proteger la información sensible durante la transmisión y almacenamiento.

4. Integrar el acceso a S3 desde el código del proyecto:

- Utilizar la biblioteca Boto3 en Python para interactuar con el bucket de S3 directamente desde el código. Esto permitirá que el equipo descargue, suba o procese las imágenes de manera dinámica durante la ejecución de los scripts de procesamiento de datos o entrenamiento de los modelos.

- `import boto3 s3 = boto3.client('s3') bucket_name = 'nombre-del-bucket'`
- `# Ejemplo para descargar una imagen desde el bucket s3.download_file(bucket_name, 'nombre_de_la_imagen.jpg', 'ruta_local`

5. Monitorización y optimización:

- Usar servicios adicionales de AWS como CloudWatch para monitorear el uso del bucket y recibir alertas en caso de cualquier actividad inusual.

- Configurar el versionado de objetos en el bucket de S3 para llevar un registro de las diferentes versiones de las imágenes en caso de que se requiera recuperar una versión anterior.

3. Extracción, limpieza y carga del conjunto de datos

El equipo ha seleccionado 4004 imágenes para realizar un análisis más eficiente. Las imágenes se han dividido equitativamente entre los miembros del equipo para su clasificación, lo que garantiza un procesamiento eficiente y uniforme de los datos. Durante esta etapa, se recortaron manualmente las imágenes para centrarse en las camas de las vacas, etiquetándolas en tres categorías: cama vacía, vaca parada y vaca acostada. Esta limpieza y preparación de los datos asegura que el modelo solo utilice la información relevante, evitando el procesamiento innecesario de áreas de la imagen que no aportan valor al análisis.

Para realizar la clasificación mencionada, hicimos uso de un script de python, el cual recorta las imágenes y nos permite eficientar el proceso de clasificación mostrando la imagen y permitiendo escoger una carpeta específica de cada clase en la cual se guardará la imagen recortada. El script puede ser consultado aquí en la carpeta de source.

Adicionalmente para nuestro proyecto, tenemos considerado implementar un modelo para detectar las vacas en las imágenes de manera automática, en lugar de hacerlo de manera manual. Para este modelo de object detection, necesitamos utilizar otro dataset que tenga las etiquetas de los diferentes bounding boxes de las vacas.

Para hacer de este proceso más eficiente (en cuestión de costo), utilizamos la herramienta Robo Flow, que es un software que utiliza inteligencia artificial para identificar las vacas y las encierra en bounding boxes. Utilizamos esta herramienta porque nos iba a salir más barato que etiquetar estas imágenes de manera manual.

4. Separación del conjunto de datos (Esquema k-fold cross validation)

El k-fold cross-validation es una técnica de validación utilizada en machine learning para evaluar el rendimiento de un modelo de manera más robusta y confiable. Sirve para garantizar que el modelo no esté sobreajustado (overfitting) ni subajustado (underfitting) y que generalice bien en datos nuevos.

Se utiliza una división en k partes: El dataset se divide en k subconjuntos (folds) del mismo tamaño y utiliza un proceso iterativo:

- En cada iteración, se entrena el modelo utilizando k-1 folds como conjunto de entrenamiento.
- El fold restante (que no se usó en el entrenamiento) se utiliza como conjunto de validación para evaluar el modelo.
- Este proceso se repite k veces, rotando los folds para que cada subconjunto se utilice una vez como validación.

Promedio de resultados: Después de realizar las k iteraciones, se calculan métricas como la precisión, el error, etc., promediando los resultados obtenidos en cada fold. Esto proporciona una estimación más confiable del rendimiento del modelo.

El script que nosotros utilizamos para hacer el k-fold cross validation se encuentra en nuestro repositorio de Github en la carpeta de source. El split del dataset se encuentra en la carpeta de dataset.

Diagrama de despliegue:

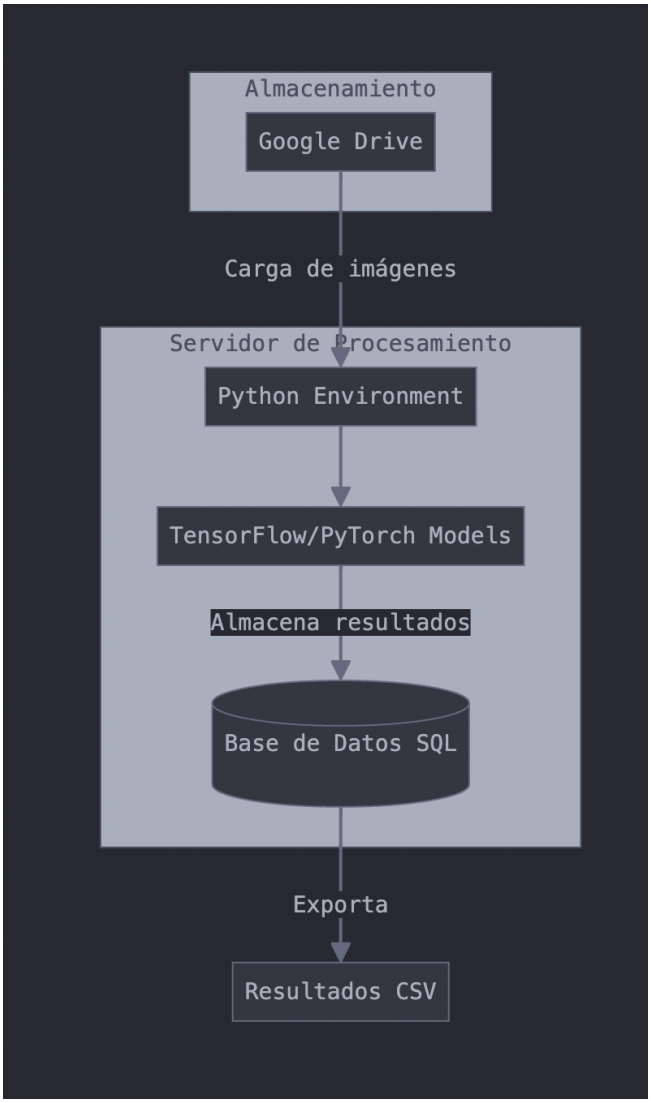
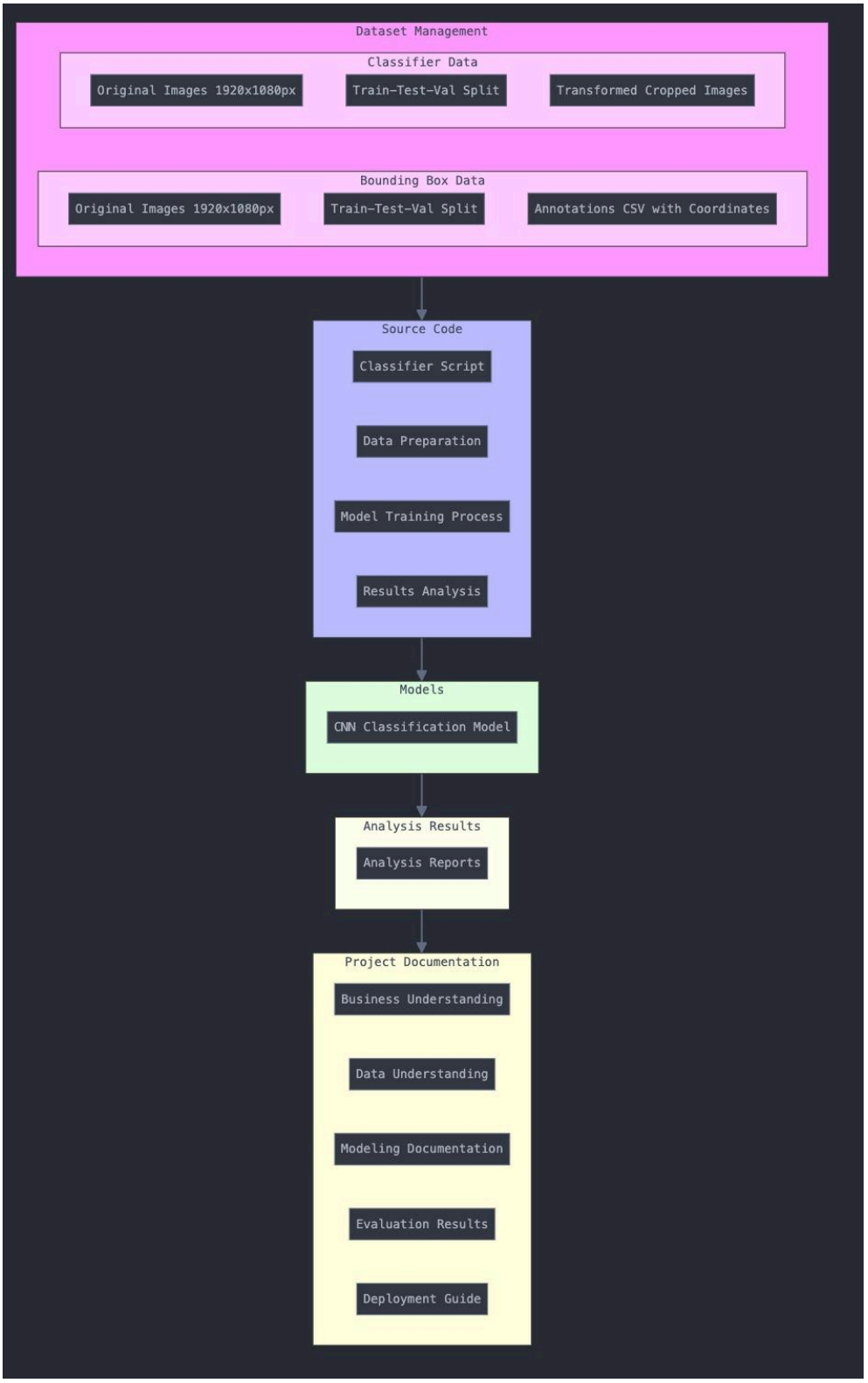


Diagrama de paquetes:



5. **¿Es necesario utilizar un enfoque orientado a Big Data? (Justificación)**

En este proyecto específico, el conjunto de datos consta de 9634 imágenes. Aunque este volumen de datos no justifica por sí solo un enfoque orientado a Big Data, el uso de tecnologías avanzadas puede ser relevante si el proyecto se expande. Si en el futuro se incluyen datos adicionales, como videos o sensores en tiempo real, la cantidad de información podría aumentar significativamente, requiriendo un procesamiento distribuido. En tal caso, herramientas como PySpark y Hadoop permitirían manejar este incremento de volumen y facilitar el análisis paralelo de grandes cantidades de datos.

Además del volumen de imágenes que tenemos, otros principios del Big Data deben considerarse:

- **Variedad**: Actualmente, el proyecto se enfoca en imágenes, pero al escalar podría incorporar otros tipos de datos (p.ej., datos de sensores, registros de actividad). Esta variedad de formatos requeriría sistemas que puedan integrar y procesar diferentes fuentes de datos de manera eficiente.
- **Velocidad**: La frecuencia de generación de datos en este proyecto es baja, ya que solo se procesan imágenes estáticas y no se necesita hacer en tiempo real. Sin embargo, si se añadieran datos en tiempo real, como flujos de video o sensores de actividad de las vacas, la velocidad de procesamiento se volvería crítica. Un enfoque de Big Data permitiría procesar estos datos en tiempo real, mejorando la capacidad de respuesta y toma de decisiones.
- **Veracidad**: La precisión y calidad de los datos es clave en el análisis de patrones de comportamiento en el rancho. Las Tecnologías de Big Data pueden ayudar a detectar inconsistencias o anomalías en datos masivos, garantizando que las conclusiones obtenidas sean confiables. Sin embargo, debido a que no son muchos datos, los revisamos manualmente en el etiquetado que se realizó. También, nos percatamos de la confiabilidad y calidad de los datos ya que estas fotos fueron tomadas en el campo de trabajo, que es donde queremos que nuestro modelo trabaje de manera correcta.
- **Valor**: El valor del análisis en este proyecto radica en optimizar el uso de camas para las vacas y mejorar la gestión de recursos. Aunque el dataset actual es manejable, un enfoque de Big Data podría potenciar el análisis y descubrir patrones a gran escala (por ejemplo si tuviéramos cámaras en muchas camas y tuviéramos que crear modelos inteligentes de todas las camas), proporcionando insights valiosos para mejorar la eficiencia operativa y el bienestar animal en el rancho.

En conclusión, aunque el proyecto actual no necesita un enfoque de Big Data debido al volumen de datos, su potencial expansión sí podría justificar la adopción de estas tecnologías para manejar la variedad, velocidad y veracidad de datos adicionales.

Anexos

Link del repositorio de GitHub: https://github.com/JP-coder2000/cattle-segmentation