Data Preparation

Adrián Matute, Pablo Martínez, Osvaldo Del Valle, Andres Callealta, Jorge Martínez

Tecnológico de Monterrey Campus Querétaro, México

1. Descripción de los datos

- **Formato**: Imágenes en formato JPG, con nombres de archivo que contienen la fecha y hora exacta de captura.
- **Cantidad**: 8115 fotos, cada una representando distintos momentos del día, capturadas en intervalos de cinco minutos. Lo que equivale a 56,4 días de captura.
- **Características descubiertas**: Condiciones extremas de luminosidad y oscuridad. Grupos de vacas con formas complejas de diferenciar.
- **Cualidad intrínseca de los datos:** Cada imagen mide 1920 por 1080 píxeles, pesando unos 0.23 MB, y tienen una profundidad de color (RGB) de 24 bits.

2. Selección de Datos

Siguiendo los objetivos de negocio, “Identificar patrones significativos de aglomeraciones en la fila de espera para el ordeño.” e “Identificar los cuellos de botella en la fila de ordeño.” Se decidió utilizar las imágenes para la tarea de detección de objetos y por ende poder tener un conteo de las vacas por cada foto.

Dadas las condiciones de iluminación de las imágenes y el error humano y ruido que estas aportarían al modelo se decidió descartar las imágenes que:

- Están sobreexpuestas, donde el exceso de luz impide identificar detalles importantes, como la posición de las vacas.
- Están subexpuestas o muy oscuras y carecen de información necesaria para detectar correctamente a las vacas.

3. Limpieza de datos

Se realizó un análisis exploratorio de los datos para encontrar el por medio de un umbral del brillo promedio de los píxeles de la imagen. Estos umbrales se decidieron observando el dataset y determinando fotos donde la iluminación deja irreconocible a los objetos en la imagen. Esto nos llevó a descartar imágenes con valores de iluminación menores a 5 unidades y mayores a 240, para un valor de 8 bits que va entre 0 a 255.

Figura 1. Foto sobreexpuesta. Brillo: 242.3263

Figura 2. Foto subexpuesta. Brillo 3.2635

Por medio de un script de python VacasLauPixels.py se realizó la eliminación de las imágenes fuera de estos rangos; este proceso nos dejó

con 8,087 imágenes, removiendo 28 las cuales representan el 0.34% de los datos originales (8,115). Se determinó que esta reducción no presenta un riesgo para el entrenamiento del modelo y por lo tanto podemos continuar construyendo el dataset.

4. Construcción del dataset

Construimos un dataset para detección de objetos múltiples en una imagen por medio de bounding boxes, cajas que engloban a objetos de una clase específica. Este dataset consta de un archivo de texto por imagen que reporta cada una de las cajas en el formato: *clase centro_x centro_y ancho alto* . Siendo la clase 0 (cow) la única que vamos a detectar y los valores de coordenadas y dimensiones números normalizados, valores entre 0 y 1, como se muestra en la Figura 3.

Figura 3. formato de bounding boxes ***ultralytics yolo format***

Se utilizó la plataforma de Roboflow para el etiquetado de las 8115 fotos (de las cuales se removieron 28 posteriormente) para formar el dataset general. Cabe recalcar que se preservó el nombre original de las imágenes ya que este representa la fecha y hora de captura y nos permitirá más adelante utilizar este atributo derivado.

Los datos se separaron en dos conjuntos, entrenamiento y validación, con una proporción de 80%, 20% respectivamente. Dejándonos con 6469 imágenes con etiquetas para entrenamiento y 1618 para validación. Antes de separar cada conjunto se aleatorizaron las fotos para garantizar la variabilidad en los datos y eliminar posibles sesgos de tiempo. Se decidió no crear un subconjunto de prueba ya que se determinó que reducir la cantidad de datos en entrenamiento y/o validación afectaría la robustez del modelo.

5. Integración de Datos

Los datos actuales únicamente vienen de la fila de ordeño de uno de los corrales y con el mismo ángulo de visión. Esto puede limitar los resultados del entrenamiento a este contexto y limitar su uso sin tener que crear un nuevo dataset y re-entrenar con este para poder aplicarlo en condiciones diferentes.

Incorporar más datos permitiría al modelo captar más variaciones y ser más adaptable a distintos escenarios, reduciendo el riesgo de overfitting y haciéndolo más robusto. Por eso, fusionar datasets de diferentes fuentes ayudaría a crear un modelo más complejo. El problema es que no se encuentran datasets etiquetados con vacas.

6. Formateo de Datos

Se decidió no cambiar el formato o tamaño de las imágenes iniciales en el dataset y por el contrario hacerlo durante el entrenamiento para conservar el máximo detalle posible para los distintos tamaños que puedan requerir las arquitecturas a evaluar.