



Tecnológico de Monterrey

Inteligencia artificial avanzada para la ciencia de datos 2

Gpo 501

Docentes

Dr. Benjamín Valdés Aguirre

Ma. Eduardo Daniel Juárez Pineda

Dr. Ismael Solis Moreno

Dr. José Antonio Cantoral Ceballos

Dr. Carlos Alberto Dorantes Dosamantes

Integrantes

Carlos Rodrigo Salguero Alcántara	A00833341
Diego Perdomo Salcedo	A01709150
Dafne Fernández Hernández	A01369230
José Emiliano Riosmena Castañón	A01704245
Luis Arturo Rendón Iñarritu	A01703572

1.0 Introducción	2
1.1 Propósito	3
2.0 Diseño de pruebas	3
2.1 División del Dataset	3
3.0 Problema, técnicas de modelado, supuestos	4
3.1 Técnica utilizada	4
3.2 Razonamiento	4
3.3 Consideraciones de usar otras tecnicas	5
3.4 Supuestos:	5
4.0 Modelado	6
4.1 Descripción/Arquitectura del modelo	6
4.2 Capa de Aumentación de Datos	7
4.2.1 Preprocesamiento de imágenes	7
4.2.2 Técnicas de Aumentación	7
4.3 Pipeline de Procesamiento	7
4.4 Hiperparámetros	7
5.0 Resultados del Modelo	8
5.1 Métricas de Entrenamiento	8
5.1.1 Precisión	8
5.1.2 Errores	8
5.2 Métricas de Prueba	8
5.3 Análisis de la Matriz de Confusión	9
5.3.1 Fortalezas	9
5.3.2 Áreas de Mejora	9
5.4 Observaciones Clave	10
5.4.1 Aspectos Positivos	10
5.4.2 Limitaciones	10
6.0 Evaluando el Modelo	10
6.1 Fortalezas del Modelo	10
6.2 Áreas de Mejora	10

1.0 Introducción

Este documento describe la tercera iteración de la fase de modelado del proyecto. La implementación de un modelo basado en XGBoost para el conteo de vacas en imágenes diurnas, aprovechando un enfoque de ensamble con extracción de características con ResNet50V2. Se utilizará el mismo dataset usado en la segunda iteración de la fase de preparación de los datos.

1.1 Propósito

El propósito principal de este documento es establecer y documentar el proceso de selección, implementación y evaluación de técnicas de modelado para lograr una mejor detección y conteo vacas que el modelo anterior. Con el proposito de lograr el objetivo de minería de datos.

Este documento sirve como guía técnica y referencia para el equipo de desarrollo, stakeholders y futuros mantenedores del sistema, asegurando la transparencia en el proceso de desarrollo y facilitando la comprensión de las decisiones técnicas tomadas durante el proyecto.

2.0 Diseño de pruebas

Para evaluar la calidad del modelo en esta tercera iteración, hemos implementado una estrategia de división temporal del dataset. Este enfoque nos permite simular condiciones reales de implementación y validar adecuadamente la precisión del modelo respecto a nuestro objetivo de minería de datos.

2.1 División del Dataset

La división del dataset se realiza siguiendo un criterio temporal, donde el 80% de los datos más antiguos se utilizan para entrenamiento y el 20% más reciente para pruebas. Esta decisión de diseño se fundamenta en la necesidad de evaluar el rendimiento del modelo en condiciones que simulan un escenario de producción real.

2.2 Validación del diseño de pruebas con objetivos de minería de datos

Esta estructura de división nos permite validar efectivamente nuestro objetivo principal de minería de datos:

Objetivo de Minería de datos:

- ❖ Determinar la cantidad de vacas en cada imagen en cualquier condición.
 - Un modelo para condiciones diurnas con un 80% de precisión.
 - Un modelo para condiciones nocturnas con un 50% de precisión.

El diseño de pruebas actual, con su distribución equilibrada y volumen expandido de datos, nos permite:

- ❖ Evaluar el rendimiento del modelo en ambas condiciones de iluminación
- ❖ Validar el cumplimiento de los umbrales de precisión establecidos

Esta estructura mejorada de validación, junto con el mayor volumen de datos, proporciona una base más sólida para evaluar el rendimiento del modelo.

3.0 Problema, técnicas de modelado, supuestos

Nuestro problema sigue siendo la clasificación y detección de vacas en imágenes, pero en esta tercera iteración nos enfocamos en mejorar la precisión mediante un enfoque híbrido que combina deep learning para extracción de características con modelos de ensamble para la clasificación final.

Las técnicas principales consideradas para esta iteración incluyen:

- Redes neuronales pre-entrenadas como extractores de características
- Modelos de ensamble (XGBoost, LightGBM, CatBoost)
- Técnicas de boosting y bagging
- Mecanismos de atención espacial
- Estrategias de pooling avanzadas

3.1 Técnica utilizada

Para esta iteración se usó un enfoque híbrido que combina dos tecnologías distintas de aprendizaje automático: **ResNet50V2** para extracción de características y **XGBoost** para clasificación. ResNet50V2 es una arquitectura de red neuronal profunda que utiliza conexiones residuales para facilitar el entrenamiento de redes muy profundas, mientras que XGBoost es un algoritmo de boosting que construye un modelo fuerte combinando múltiples árboles de decisión débiles. Esta combinación aprovecha la capacidad de las redes neuronales

para extraer características complejas de imágenes y la eficiencia de los modelos de boosting para realizar clasificaciones precisas basadas en estas características.

3.2 Razonamiento

La elección de esta arquitectura híbrida se fundamenta en varios aspectos clave:

1. **Extracción robusta de características:**
 - ResNet50V2 probó ser efectivo en iteraciones anteriores
 - El mecanismo de atención mejora el enfoque en características relevantes
 - La estrategia de pooling dual captura mejor la información espacial
2. **Ventajas del modelo de ensamble:**
 - XGBoost proporciona mayor interpretabilidad
 - Mejor manejo de relaciones no lineales
 - Capacidad de procesamiento en GPU para mayor eficiencia
 - Mayor robustez ante overfitting

3.3 Consideraciones de usar otras técnicas

Las alternativas consideradas fueron descartadas por las siguientes razones:

1. **LightGBM:**
 - Menor estabilidad en datasets pequeños
 - Más sensible a hiperparámetros
 - Menos opciones de optimización para GPU
2. **CatBoost:**
 - Mayor tiempo de entrenamiento
 - Menor flexibilidad en la configuración
 - Recursos computacionales más intensivos
3. **Random Forest:**
 - Menor capacidad para capturar patrones complejos
 - Tiempo de inferencia más largo
 - Menor precisión en pruebas preliminares
4. **Gradient Boosting tradicional:**
 - Menor velocidad de entrenamiento
 - No aprovecha capacidades de GPU
 - Menor precisión que XGBoost

3.4 Supuestos:

Supuestos sobre el Entorno:

- ❖ Las condiciones de iluminación permiten distinguir objetos del fondo.
- ❖ No hay factores ambientales extremos (niebla densa).
- ❖ La distancia entre la cámara y los objetos está dentro del rango de entrenamiento.
- ❖ El ángulo de visión es similar al de las imágenes de entrenamiento.

Supuestos sobre los Datos de Entrada:

- ❖ Las imágenes de entrada tienen una resolución mínima suficiente para distinguir vacas (al menos 640x640 píxeles).
- ❖ Los objetos a detectar (vacas) ocupan al menos un 5% del área total de la imagen.
- ❖ Las imágenes son a color (3 canales RGB).

Supuestos sobre los Objetos:

- ❖ Las vacas aparecen en posiciones naturales (de pie, pastando, caminando).
- ❖ Los objetos son distinguibles del fondo.

Supuestos sobre las Condiciones de Operación:

- ❖ La Raspberry Pi opera dentro de su rango de temperatura normal.
- ❖ Hay suficiente memoria disponible para el procesamiento.
- ❖ La fuente de alimentación es estable y suficiente.

4.0 Modelado

En esta sección se detallarán las características específicas del modelo implementado, incluyendo su funcionamiento, parámetros, comportamiento y limitaciones. Algo importante es que para esta iteración del modelado solo lo haremos con el diurno, ya que obtuvimos buenos resultados con el nocturno en el modelo anterior.

4.1 Descripción/Arquitectura del modelo

La implementación utiliza un enfoque híbrido innovador que combina la extracción de características basada en ResNet50V2 con XGBoost como clasificador final. ResNet50V2 procesa imágenes de 224x224x3 píxeles y utiliza un mecanismo de atención espacial junto con

pooling dual para extraer características relevantes. La verdadera innovación radica en el uso de XGBoost como clasificador final, un algoritmo de gradient boosting que construye un modelo robusto a través de la combinación iterativa de múltiples árboles de decisión. Esta arquitectura aprovecha la capacidad de las redes neuronales para procesar imágenes complejas mientras utiliza la eficiencia y precisión de XGBoost para la clasificación final, permitiendo un mejor manejo de relaciones no lineales y proporcionando mayor interpretabilidad en las predicciones.

4.2 Capa de Aumentación de Datos

4.2.1 Preprocesamiento de imágenes

- ❖ Reducción de ruido: Fast Non-Local Means
- ❖ Mejora de contraste: CLAHE
- ❖ Normalización en un rango de -1 a 1

Razón: Estas técnicas de preprocesamiento se seleccionaron para mejorar la calidad de las imágenes y estandarizar la entrada antes de la extracción de características.

4.2.2 Técnicas de Aumentación

- ❖ Ajuste aleatorio de brillo ($\pm 10\%$)
- ❖ Ajuste aleatorio de contraste ($\pm 10\%$)
- ❖ Volteo horizontal
- ❖ Rotación pequeña (± 5 grados)

Razón: Se implementaron aumentaciones moderadas para mejorar la robustez del modelo sin distorsionar excesivamente las características esenciales de las imágenes.

4.3 Pipeline de Procesamiento

- ❖ Preprocesamiento y aumentación de imágenes
- ❖ Extracción de características via ResNet50V2
- ❖ Normalización de características
- ❖ Predicción ensemble XGBoost

Razón: El pipeline se diseñó para maximizar la eficacia del enfoque híbrido, asegurando una transición suave desde el preprocesamiento hasta la clasificación final.

4.4 Hiperparámetros

- ❖ Número de estimadores: 200
- ❖ Learning rate: 0.03
- ❖ Max depth: 0.6
- ❖ Subsample: 0.8
- ❖ Colsample_bytree: 0.8
- ❖ Tree method: gpu hist
- ❖ Objective: reg:squarederror

Razón: Los hiperparámetros se ajustaron para optimizar el rendimiento de XGBoost en GPU mientras se mantiene un balance entre velocidad de entrenamiento y capacidad de generalización.

5.0 Resultados del Modelo

5.1 Métricas de Entrenamiento

5.1.1 Precisión

- ❖ Exactitud precisa: 45.02%
- ❖ Precisión ± 1 vaca: 91.58%
- ❖ Precisión ± 2 vacas: 99.40%

5.1.2 Errores

- ❖ Error Cuadrático Medio (MSE): 0.7461
- ❖ Error Absoluto Medio (MAE): 0.6783
- ❖ Raíz del Error Cuadrático Medio (RMSE): 0.8637

5.2 Métricas de Prueba

- ❖ MSE: 1.8548
- ❖ MAE: 1.0529
- ❖ RMSE: 1.3619

5.3 Análisis de la Matriz de Confusión

Confusion Matrix

0	31	40	24	7	3	0	0	0	0	0	0	0	0
1	2	21	82	67	12	1	1	1	0	0	0	0	0
2	1	7	52	117	58	8	2	0	0	0	0	0	0
3	0	0	30	107	111	43	12	0	0	0	0	0	0
4	0	0	12	67	97	90	44	17	1	0	0	0	0
5	0	0	6	18	73	96	66	30	8	1	0	0	0
6	0	0	1	6	18	52	54	62	31	1	0	0	0
7	0	0	0	2	1	11	27	31	26	11	0	0	0
8	0	0	0	1	0	1	6	17	16	15	0	0	0
9	0	0	0	0	0	1	1	6	13	10	0	0	0
10	0	0	0	0	0	0	0	1	6	3	0	0	0
11	0	0	0	0	0	0	0	2	3	5	0	0	0
12	0	0	0	0	0	0	0	0	0	1	0	0	0
	0	1	2	3	4	5	6	7	8	9	10	11	12

Predicted

5.3.1 Fortalezas

- ❖ Excelente rendimiento en el rango de 2-4 vacas
- ❖ Muy buena predicción para grupos pequeños (0-3 vacas)
- ❖ Alta concentración de predicciones en la diagonal principal

5.3.2 Áreas de Mejora

- ❖ Mayor dispersión en conteos altos (>7 vacas)
- ❖ Tendencia a subestimar grupos grandes
- ❖ Menor precisión en los extremos del rango

5.4 Observaciones Clave

5.4.1 Aspectos Positivos

- ❖ El modelo es extremadamente preciso dentro de un margen de ± 1 vaca (91.58%)
- ❖ Casi perfecto dentro de ± 2 vacas (99.40%)
- ❖ Rendimiento constante en el rango medio de conteos

5.4.2 Limitaciones

- ❖ Diferencia notable entre métricas de entrenamiento y prueba
 - MSE: 0.7461 vs 1.8548
 - MAE: 0.6783 vs 1.0529
- ❖ Mayor dificultad para predecir grupos grandes de vacas
- ❖ Algunas confusiones entre categorías adyacentes.

5.5 Limitaciones y Consideraciones

Categoría	Limitación	Justificación
Precisión General	Exactitud limitada en conteo exacto	Precisión exacta de 45.02%, aunque mejora significativamente con margen de error
Grupos Grandes	Dificultad en conteos altos	Mayor dispersión y tendencia a subestimar grupos de >7 vacas
Rendimiento en Pruebas	Degradación de métricas	MSE aumenta de 0.7461 a 1.8548, MAE de 0.6783 a 1.0529
Consistencia	Variabilidad entre categorías	Excelente en 2-4 vacas, menor precisión en extremos del rango
Predicción	Sesgo en subestimación	Tendencia a subestimar el conteo en grupos grandes
Generalización	Gap entre entrenamiento y prueba	Diferencia significativa en todas las métricas de error

6.0 Evaluando el Modelo

La tercera iteración, que implementó un enfoque híbrido con ResNet50V2 y XGBoost, mostró los siguientes resultados:

Precisión del modelo:

- Precisión exacta: 45.02%
- Precisión con margen de ± 1 vaca: 91.58%
- Precisión con margen de ± 2 vacas: 99.40%

Métricas de error en entrenamiento:

- MSE: 0.7461
- MAE: 0.6783
- RMSE: 0.8637

Métricas de error en pruebas:

- MSE: 1.8548
- MAE: 1.0529
- RMSE: 1.3619

6.1 Validación de Objetivos

NO LOGRAMOS EL OBJETIVO

Objetivo de Minería de Datos

- ❖ Determinar la cantidad de vacas en cada imagen en cualquier condición.
 - Un modelo para condiciones diurnas con un 80% de precisión.
 - Un modelo para condiciones nocturnas con un 50% de precisión.

La tercera iteración mostró resultados que, aunque mejoraron respecto a las iteraciones anteriores, aún no alcanzaron completamente los objetivos establecidos. La precisión exacta del 45.02% queda por debajo del objetivo del 80% para condiciones diurnas, sin embargo, el modelo mostró una capacidad notable para predicciones aproximadas, con un 91.58% de precisión dentro de un margen de ± 1 vaca y un impresionante 99.40% dentro de ± 2 vacas. Esta discrepancia entre la precisión exacta y la precisión con márgenes sugiere que el enfoque híbrido de ResNet50V2 con XGBoost, aunque innovador, todavía enfrenta desafíos

para lograr el nivel de precisión exacta deseado. Las métricas de error (MSE: 0.7461 vs 1.8548 en pruebas) indican una brecha significativa entre el rendimiento de entrenamiento y prueba, sugiriendo que aún hay espacio para mejoras en la capacidad de generalización del modelo, particularmente en el conteo de grupos grandes de vacas.

6.1 Sigüientes pasos

Aunque no alcanzamos el objetivo del 80% de precisión exacta, el modelo mostró una capacidad notable para predicciones aproximadas, siendo extremadamente preciso cuando se permite un margen de error de ± 1 o ± 2 vacas. Vamos a probar con otras técnicas de modelado para intentar alcanzar nuestro objetivo.