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

Querétaro, Querétaro

1.0 Modelado **4**
1.1 Introducción 4
1.2 Propósito 4
2.0 Diseño de las pruebas **4**
2.1 Pruebas del modelo 4
2.2 Separación del Dataset 5
2.3 Validación del diseño de pruebas con objetivos de minería de datos 5
3.0 Problema, técnica de modelado, supuestos **6**
3.1 Técnica utilizada 6
3.2 Razonamiento 7
3.3 Consideraciones de usar otras técnicas 7
3.4 Supuestos: 8
4.0 Modelado **9**
4.1 Descripción/Arquitectura del modelo 9
4.1.1 Arquitectura y Funcionamiento 9
4.2 Proceso de detección 10
4.2.1 Preprocesamiento 10
4.2.2 Forward Pass 10
4.2.3 Post Procesamiento 10
4.3 Optimizaciones específicas 10
4.4 Lista de los parámetros 11
4.4.1 Modelo base 11
4.4.2 Configuración de entrenamiento 11
4.4.3 Augmentations 11
4.4.4 Configuración de detección 12
5.0 Modelo resultante **12**
5.1 Comportamiento 12
5.1.1 R-Curve 14
5.1.2 P-Curve 15
5.1.3 F1-Curve 16
5.1.4 Matriz de Confusión 17
5.1.5 Matriz de Confusión Normalizada 18
5.2 Limitaciones y Consideraciones 19
6.0 Evaluar los resultados **20**
6.1 Resultados de las pruebas 20
6.2 Análisis de la Matriz de Confusión 20
6.3 Validación de Objetivos 20
6.4 Impacto en Términos de Negocio 21
6.4.1 Hallazgos 21
7.0 Refinamiento **23**
7.1 Cambios realizados 23
7.1.1 Análisis de Curvas de Rendimiento 23
7.1.2 Evolución del Entrenamiento 25
7.2 Resultados 26

7.3 Mantenimiento de Calidad 26
7.4 Conclusiones del Refinamiento 26
8.0 Prueba de arquitectura **26**
8.1 Hardware 26
8.2 Software 27
8.3 Nube 27
8.4 Videos Prueba de Arquitectura 27

1.0 Modelado

Este documento contiene la descripción del modelo benchmark y su refinamiento.

1.1 Introducción

Además describe el proceso de desarrollo, implementación y evaluación de un sistema de detección y conteo automatizado de ganado bovino mediante técnicas de visión computacional. El sistema desarrollado emplea técnicas de deep learning, específicamente el modelo YOLOv8x (You Only Live Once versión 8 extra large), para detectar y contar vacas en imágenes capturadas bajo diversas condiciones de iluminación.

1.2 Propósito

El propósito principal de este documento es establecer y documentar el proceso de selección, implementación y evaluación de técnicas de modelado para lograr la detección y conteo preciso de vacas en imágenes de ganado bovino.

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 las pruebas

Para realizar el modelado, primero diseñamos las pruebas considerando la naturaleza de nuestro conjunto de datos de imágenes de vacas. El conjunto de datos se estructura para permitir la detección de objetos (vacas) en las imágenes mediante el modelo YOLO.

2.1 Pruebas del modelo

Para evaluar la calidad del modelo, dividimos el dataset en dos conjuntos principales: entrenamiento y validación. Esta división nos permite entrenar el modelo y validar su precisión de manera independiente. La distribución se realizó con aproximadamente 81% para entrenamiento (7,283 imágenes) y 19% para validación (1,658 imágenes), manteniendo una proporción que asegura suficientes datos para ambas fases.

2.2 Separación del Dataset

La división del dataset se realizó manteniendo una estructura organizada:

- Entrenamiento: 7,283 imágenes
  - 6,469 imágenes con vacas
  - 1,218 imágenes de fondo
- Validación: 1,658 imágenes
  - 1,618 imágenes con vacas
  - 144 imágenes de fondo

Esta distribución asegura que tanto el conjunto de entrenamiento como el de validación contengan una mezcla representativa de imágenes con y sin vacas, permitiendo una evaluación robusta del rendimiento del modelo.

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

El diseño de pruebas utiliza cuatro métricas fundamentales para validar el cumplimiento de nuestro objetivo 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.

Métricas:

- **Precisión (P):**
  - Mide el porcentaje de detecciones correctas entre todas las detecciones realizadas
  - Nos ayuda a validar la exactitud de las predicciones del modelo
  - Directamente relacionada con nuestros criterios de éxito de 80% para condiciones diurnas y 50% para nocturnas
  - Un valor alto indica que el modelo tiene pocos falsos positivos
- **Recall (R):**
  - Indica la proporción de vacas reales que el modelo logra detectar
  - Crucial para asegurar que no se están omitiendo vacas en el conteo
  - Complementa la precisión para una evaluación más completa

- ➢ Un valor alto significa que el modelo raramente pasa por alto vacas presentes

❖ **mAP50 (Mean Average Precision con IoU=50%):**

- ➢ Evalúa la precisión del modelo considerando un umbral de solapamiento del 50%
- ➢ Métrica estándar para evaluar la calidad de las detecciones
- ➢ Proporciona una visión general del rendimiento del modelo
- ➢ Útil para comparar con los umbrales de precisión establecidos en nuestros criterios

❖ **mAP50-95:**

- ➢ Promedio de mAP en diferentes umbrales de IoU (de 50% a 95%)
- ➢ Evalúa la precisión de la localización de las detecciones
- ➢ Indica qué tan preciso es el modelo en diferentes niveles de exigencia
- ➢ Ayuda a validar la robustez general del modelo

Utilizando estas métricas podremos validar el objetivo de minería de datos adecuadamente. A pesar de que nuestro objetivo inicial era 50% de precisión con un modelo nocturno y 80% con un modelo diurno, con esto podremos cumplir con ambos a la vez, ya que el dataset utilizado contiene imágenes con ambos tipos de iluminación y se realizará un modelo generalizado.

3.0 Problema, técnica de modelado, supuestos

Nuestro problema es de detección de objetos, una subcategoría específica de la clasificación.

Las técnicas principales para esto incluyen:

- Redes neuronales convolucionales (CNN)
- Arquitecturas basadas en regiones (R-CNN, Fast R-CNN)
- Single Shot Detectors (SSD)
- YOLO (You Only Look Once)
- Cascade Detectors
- Feature Pyramid Networks (FPN)

3.1 Técnica utilizada

Para nuestro se seleccionó **YOLO versión 8x**, que representa una variante más potente y sofisticada de la familia de detectores YOLO. Yolov8x utiliza una arquitectura de red

neuronal convolucional profunda con aproximadamente 68.2 M de parámetros, diseñada para maximizar la precisión en tareas de detección de objetos, ofreciendo un rendimiento en términos de exactitud y confiabilidad.

3.2 Razonamiento

La elección de YOLO como técnica de modelado se fundamenta en varios aspectos clave relacionados con nuestro problema específico de detección de vacas:

1. **Naturaleza del Objeto a Detectar:**
    - Las vacas son objetos de tamaño relativamente grande y consistente en las imágenes
    - Presentan características distintivas y patrones reconocibles
    - Pueden aparecer en diferentes posturas y orientaciones
    - Suelen estar en grupos, requiriendo detección múltiple
2. **Condiciones de Operación:**
    - Necesidad de procesamiento en tiempo real para monitoreo continuo
    - Implementación en hardware limitado (Raspberry Pi)
    - Operación en entornos externos con condiciones variables
    - Requisito de respuesta rápida para aplicaciones de monitoreo
3. **Ventajas Técnicas de YOLO:**
    - Procesamiento en una sola pasada, ideal para detección en tiempo real
    - Capacidad de detectar múltiples objetos simultáneamente
    - Buen manejo de objetos en diferentes escalas y orientaciones
    - Arquitectura eficiente que optimiza recursos computacionales

La combinación de estos factores hace que YOLO sea particularmente adecuado para nuestro caso de uso específico. Esta técnica fue seleccionada por su capacidad de procesar imágenes en tiempo real mientras mantiene una alta precisión en la detección de objetos.

3.3 Consideraciones de usar otras técnicas

Durante la fase de selección del modelo, se evaluaron las técnicas anteriores disponibles, pero fueron descartadas por diversas razones:

1. **Redes Neuronales Convolucionales (CNN) básicas:**
    - Requieren procesamiento adicional para localización

- No están optimizadas para detección en tiempo real
- Mayor complejidad en la implementación para detección de múltiples objetos
- Descartadas por la necesidad de procesamiento en tiempo real

2. **Arquitecturas basadas en regiones (R-CNN, Fast R-CNN):**
   - Proceso de dos etapas que aumenta el tiempo de procesamiento
   - Mayor consumo de recursos computacionales
   - Requieren más memoria durante la inferencia
   - No viables para implementación en Raspberry Pi debido a limitaciones de recursos
3. **Single Shot Detectors (SSD):**
   - Menor precisión en objetos pequeños o distantes
   - Rendimiento inferior a YOLO en condiciones variables de iluminación
   - Menor capacidad para manejar objetos en diferentes escalas
   - Descartadas por requerir mayor capacidad computacional para lograr precisión similar
4. **Cascade Detectors:**
   - Complejidad computacional significativamente mayor
   - Tiempo de entrenamiento extenso
   - Dificultad para optimizar múltiples detectores en cascada
   - No adecuados para procesamiento en tiempo real en hardware limitado
5. **Feature Pyramid Networks (FPN):**
   - Mayor complejidad arquitectónica
   - Requieren más memoria durante la inferencia
   - Tiempo de procesamiento más alto
   - Beneficios no justifican el costo computacional adicional

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.

4.1 Descripción/Arquitectura del modelo

YOLOv8x es el modelo más potente de la familia de arquitecturas YOLO para detección de objetos en tiempo real. A diferencia de los enfoques tradicionales que utilizan ventanas deslizantes o propuestas de regiones, YOLO trata la detección como un problema de regresión directa, procesando la imagen completa en una sola pasada.

4.1.1 Arquitectura y Funcionamiento

El modelo divide la imagen de entrada (640x640 pixeles) en una cuadrícula y procesa toda la imagen simultáneamente a través de una red neuronal convolucional profunda. Para cada celda de la cuadrícula, el modelo predice:

- Bounding boxes (cajas delimitadores)
- Confidence scores (puntuaciones de confianza)
- Probabilidad de clase (en este caso, solo ‘cow’)

La arquitectura YOLOv8x consta de tres componentes principales:

Backbone

- Red convolucional CSPDarknet como arquitectura base
- Implementa Cross-Stage-Partial connections para mejorar el flujo de información
- Reduce dimensionalidad mientras mantiene características importantes
- Mayor capacidad de extracción de características que versiones más pequeñas

Neck

- Estructura PANet (Path Aggregation Network)
- Crea una jerarquía de características
- Permite la detección de objetos a múltiples escalas
- Facilita la propagación de información entre diferentes niveles

Head

- Realiza las predicciones finales
- Genera bounding boxes y scores de confianza
- Utiliza anchors-free detection
- Implementa decodificación directa de las predicciones

4.2 Proceso de detección

4.2.1 Preprocesamiento

- Redimensionamiento de la imagen a 640x640
- Normalización de valores de píxeles
- Aplicación de aumentaciones (durante el entrenamiento)

4.2.2 Forward Pass

- La imagen atraviesa el backbone extrayendo características
- El neck procesa y combina características a múltiples escalas
- El head genera predicciones para cada región

4.2.3 Post Procesamiento

- Filtrado de predicciones por umbral de confianza
- Aplicación de Non-Maximum Suppression (NMS)
- Generación de predicciones finales

4.3 Optimizaciones específicas

Para nuestro caso de uso específico, el modelo fue configurado con:

- Una sola clase de predicción
- Data augmentation específico para escenas ganaderas
- Early stopping para prevenir overfitting
- Sistema de tracking temporal para mejorar la consistencia

4.4 Lista de los parámetros

4.4.1 Modelo base

- Valor: YOLOv8x.pt

Razón: máxima capacidad y precisión para detección de las vacas

4.4.2 Configuración de entrenamiento

- Image Size: 640 x 640 píxeles
- Batch Size: 16
- Epochs: 50
- Device: GPU ("0")
- Patience: 20
- Dropout: 0.2

Razón: Configuración optimizada para balancear la precisión del entrenamiento con las limitaciones de hardware y memoria, permitiendo un aprendizaje efectivo sin sobreajuste.

4.4.3 Augmentations

- Mosaic: 1.0
- Mixup: 0.3
- Copy Paste: 0.3
- Degrees: 45.0
- Translate: 0.2
- Scale: 0.5
- Shear: 10.0
- Perspective: 0.0005
- Flip: 0.5 (tanto horizontal como vertical)
- HSV Adjustments:
    - Hue: 0.015
    - Saturation: 0.7
    - Value: 0.4

Razón: Combinación de técnicas de aumentación diseñadas para mejorar la robustez del modelo ante diferentes condiciones de iluminación, ángulos y posiciones de las vacas.

4.4.4 Configuración de detección

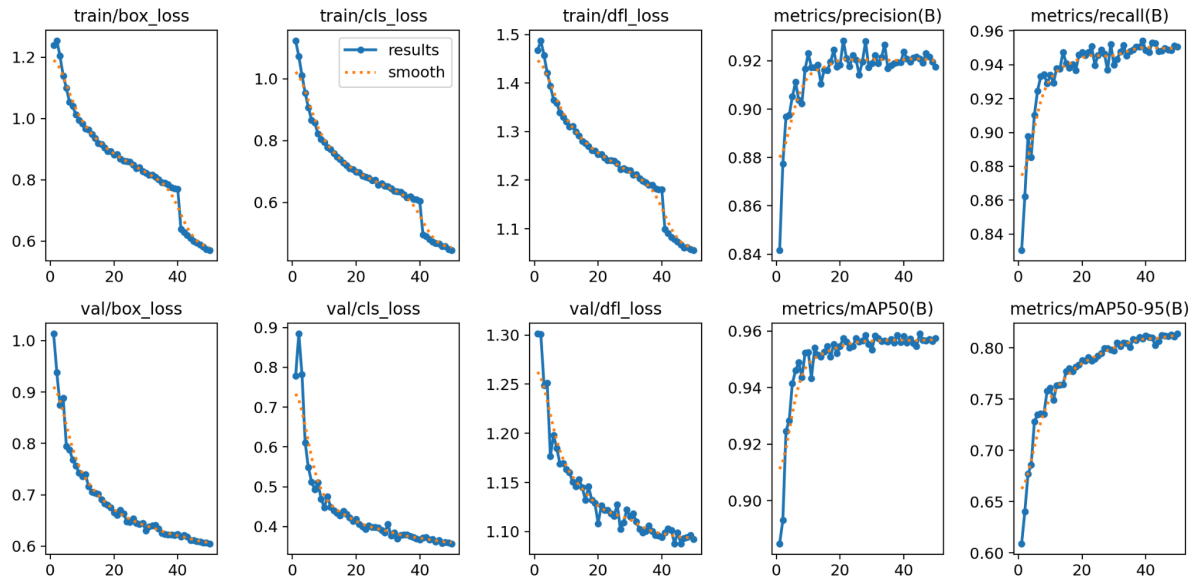
- Confidence Threshold: 0.30
- IoU Threshold: 0.60
- Minimum Area: 500
- Maximum Overlap Ratio: 0.8
- Temporal Window: 3

Razón: Parámetros ajustados para maximizar la precisión en la detección mientras se minimizan los falsos positivos y se mantiene la estabilidad temporal de las predicciones.

5.0 Modelo resultante

El modelo resultante utiliza la arquitectura YOLOv8x completa, siendo la versión más grande y potente de la familia YOLOv8x.

5.1 Comportamiento



Esta colección de gráficas muestra cómo el modelo ha ido aprendiendo y mejorando durante su entrenamiento. Se divide en dos tipos principales de métricas: las pérdidas que deben disminuir, y las métricas de rendimiento que deben aumentar.

Los resultados de pérdidas muestran:

Métrica	Descripción
Train/Val Box Loss (Pérdida en localización)	Disminuye de 1.2 a 0.56. Indica que el modelo mejoró significativamente en encontrar la ubicación exacta de los objetos. La curva suave descendente sugiere un aprendizaje estable.
Train/Val Cls Loss (Pérdida de clasificación)	Reduce de 1.1 a 0.38. Demuestra que el modelo mejoró sustancialmente en identificar correctamente qué tipo de objeto está viendo. La reducción constante indica un buen aprendizaje de características distintivas.
Train/Val DFL Loss (Pérdida en distribución)	Baja de 1.5 a 1.08. Refleja una mejora en la calidad de las predicciones de distribución. Estabilización constante indica un buen aprendizaje de características distintivas.

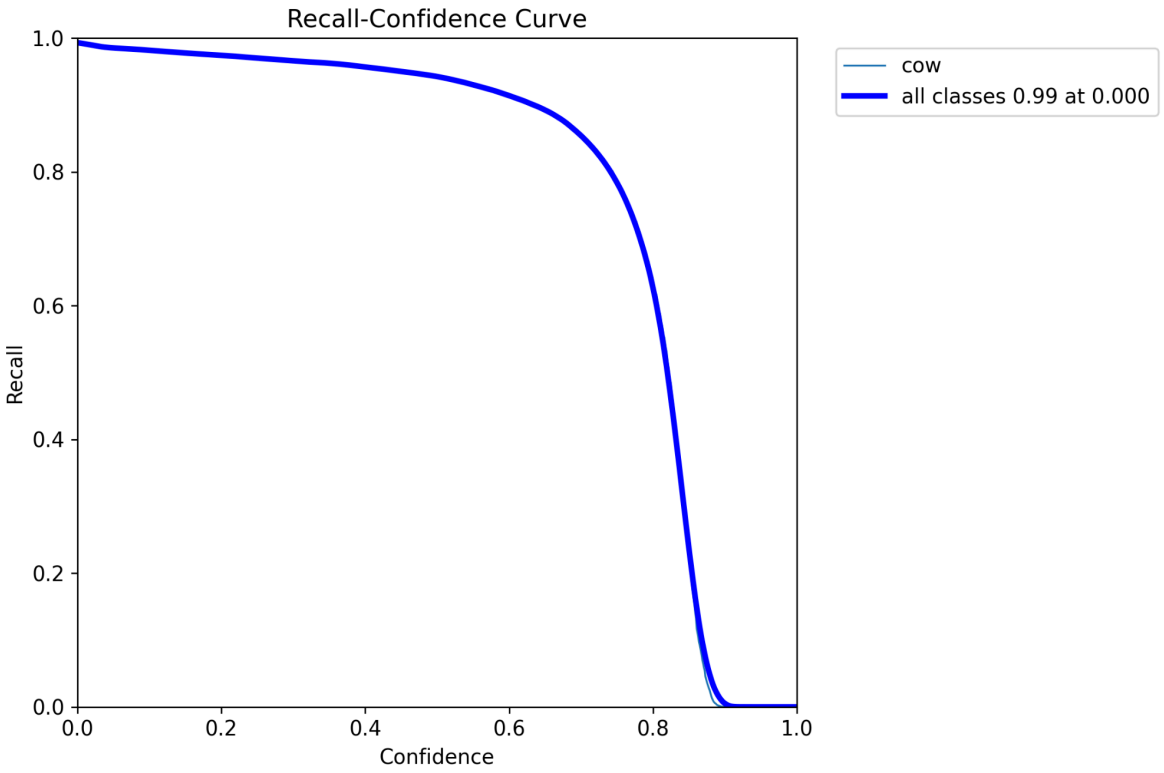
Los resultados de las métricas de rendimiento muestran:

Métrica	Descripción
Precisión (B)	Alcanza 0.92 (92%). Indica que de cada 100 detecciones que hace el modelo, 92 son correctas. La estabilidad en valores altos sugiere un aprendizaje robusto.
Recall (B)	Llega a 0.95 (95%). Significa que el modelo encuentra 95 de cada 100 objetos que debería detectar. Excelente capacidad de detección general.
mAP50 (B)	Alcanza 0.95 (95%). Representa un rendimiento sobresaliente en detecciones con 50% de superposición. Confirma la alta calidad general del modelo.
mAP50-95 (B)	Logra 0.80 (80%). Demuestra un rendimiento sólido incluso con criterios más estrictos de superposición. Indica que el modelo es preciso incluso en condiciones exigentes.

En términos prácticos, estas métricas revelan un modelo que:

- ❖ Ha aprendido de manera consistente y estable
- ❖ Tiene un rendimiento sobresaliente en todas las métricas importantes
- ❖ Está bien equilibrado entre precisión y capacidad de detección
- ❖ Mantiene su rendimiento incluso bajo criterios estrictos de evaluación
- ❖ Está listo para su uso en aplicaciones reales exigentes.

5.1.1 R-Curve



Esta gráfica muestra qué tan bueno es el modelo para encontrar todos los objetos que debería detectar. El recall nos dice, del total de objetos que existen, cuántos logra encontrar del modelo. Es importante porque nos indica si el modelo está aprendiendo objetos que debería encontrar.

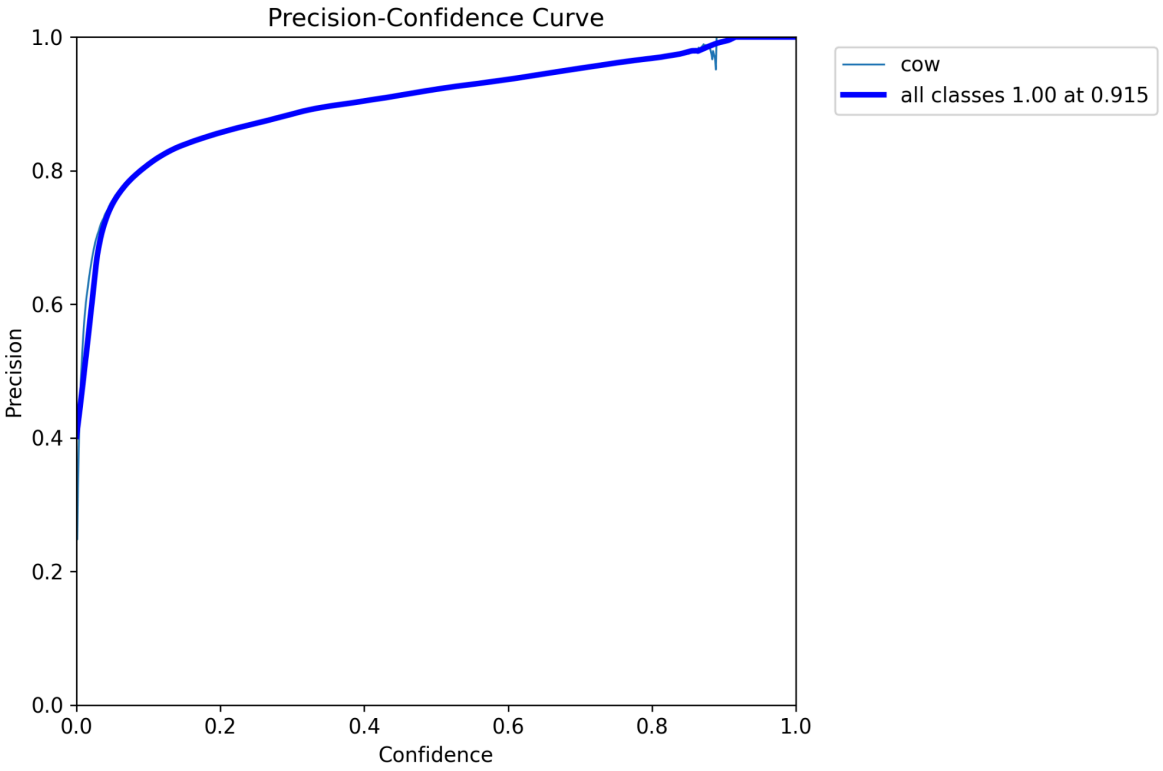
Los resultados nos muestran que

- ❖ El modelo comienza con un recall de 0.99 (99%) a baja confianza
  - ➢ Cuando el modelo está en su configuración más sensible, encuentra 99 de cada 100 objetos que debería detectar.
  - ➢ Es un resultado sobresaliente que indica una excelente capacidad de detección base.
- ❖ El rendimiento se mantiene alto hasta un umbral de confianza de 0.8

- ➢ El modelo conserva su alta capacidad de detección, incluso cuando le pedimos estar más seguro de sus predicciones.
- ➢ Después de 0.8, el recall cae significativamente porque el modelo se vuelve demasiado exigente para hacer una detección.

❖ La caída después de 0.8 es pronunciada, pero controlada, indicando el punto donde el modelo comienza a preferir la certeza sobre la cobertura.

5.1.2 P-Curve



Muestra que tan confiable es el modelo cuando dice que ha encontrado algo. La precisión nos indica qué porcentaje de las detecciones del modelo son realmente correctas, es decir, cuando el modelo dice “encontré un objeto”, qué tan seguros podemos estar de que realmente es un objeto.

Los resultados nos muestran que:

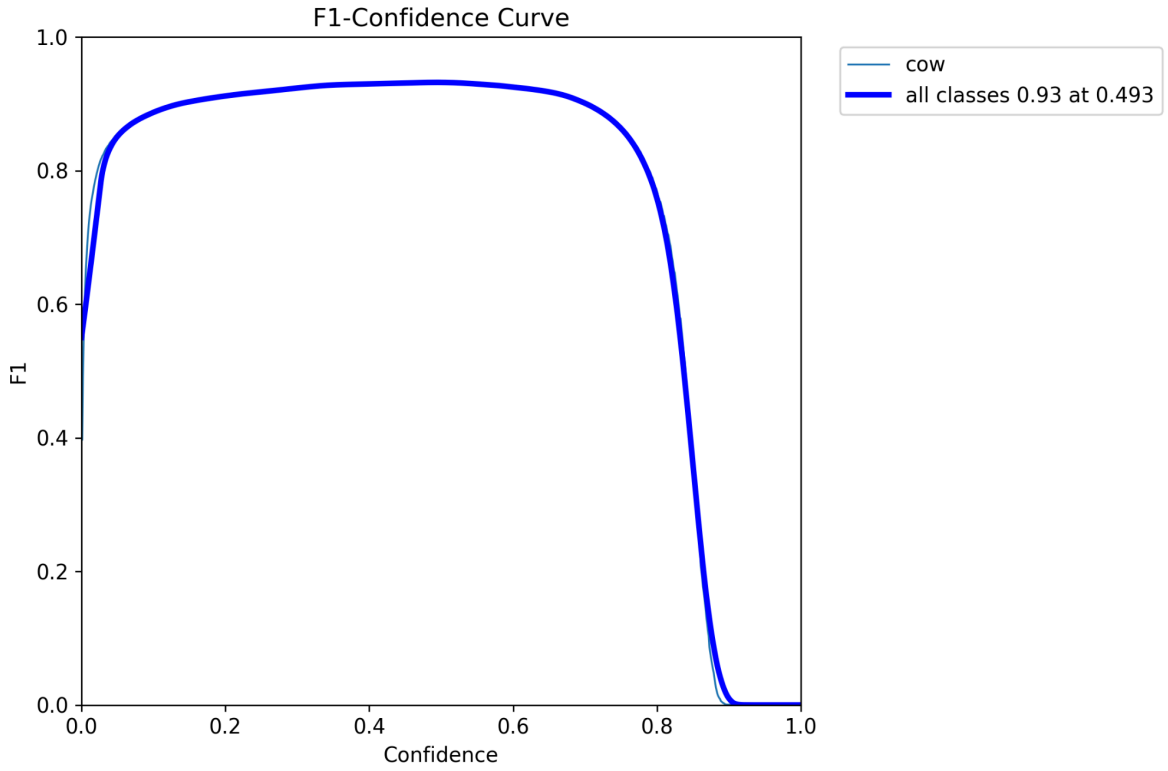
- ❖ El modelo alcanza una precisión perfecta de 1.00 (100%) con un umbral de confianza de 0.915. Significando que el modelo está muy seguro de sus detecciones.
- ❖ La curva asciende de manera constante y gradual.
  - ➢ A medida que aumentamos el umbral de confianza, el modelo se vuelve más preciso.

- ➢ No hay caídas o subidas bruscas, lo que sugiere un aprendizaje robusto y estable.

❖ Incluso a umbrales de confianza más bajos, mantiene una precisión respetable

- ➢ Esto permite flexibilidad en la configuración según las necesidades específicas de cada aplicación.
- ➢ Podemos ajustar el balance entre precisión y flexibilidad según se requiera.

5.1.3 F1-Curve



Esta gráfica es la calificación general del modelo que combina dos aspectos importantes: qué tan preciso es cuando detecta algo y cuántas cosas puede detectar del total. Un F1-score alto significa que el modelo es bueno tanto encontrando objetos como asegurándose de que sus detecciones sean correctas.

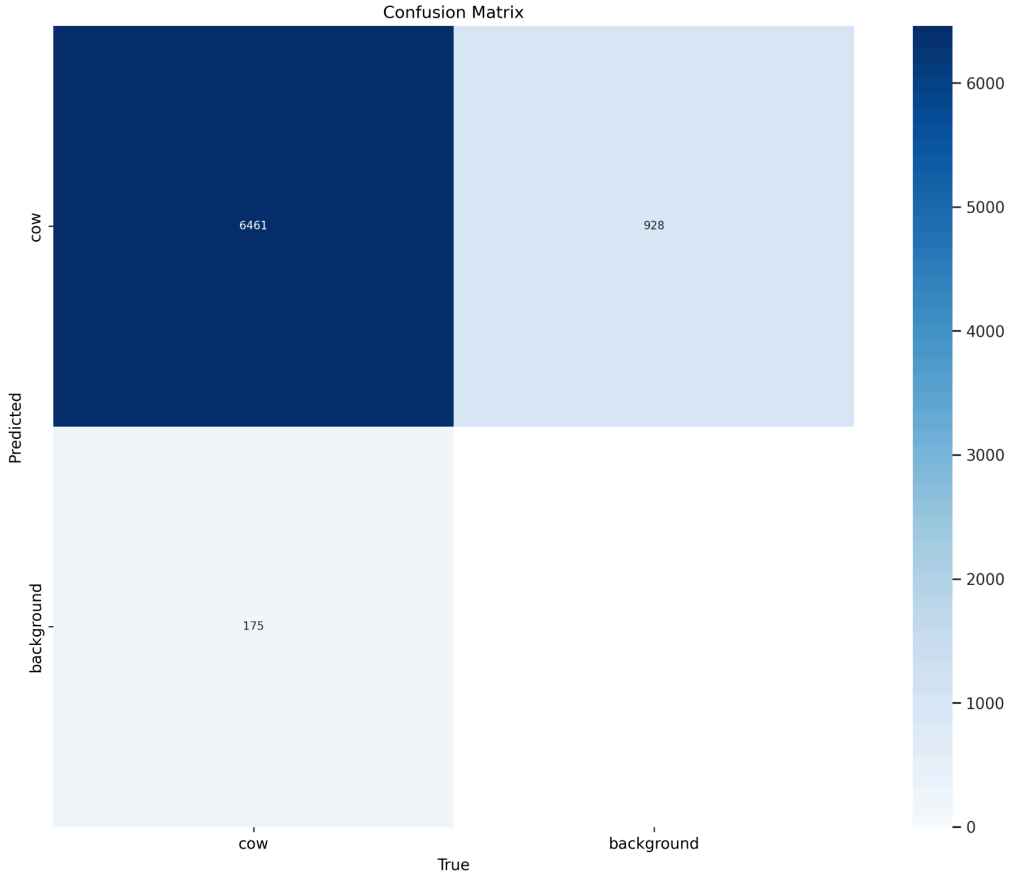
Los resultados muestran que

❖ El modelo alcanza su mejor rendimiento con un F1 de 0.93 (93%) cuando se configura con un umbral de confianza de 0.493. Esto significa que el modelo está acertando en 93 de cada 100 casos en su punto óptimo.

❖ El rendimiento se mantiene estable hasta un umbral de confianza de 0.8. Esto demuestra que el modelo es robusto y flexible en su uso.

- ➢ Adicionalmente, demuestra que es posible ajustar el nivel de confianza en un rango amplio sin perder rendimiento.
- ➢ La caída después de 0.8 es natural y esperada, ya que el modelo se vuelve demasiado selectivo.

5.1.4 Matriz de Confusión



Esta matriz muestra cómo se desempeña el modelo en sus predicciones, comparando lo que predijo contra lo que realmente era. Nos ayuda a entender los aciertos y errores del modelo en la clasificación entre vacas y el fondo (background).

Los resultados nos muestran que:

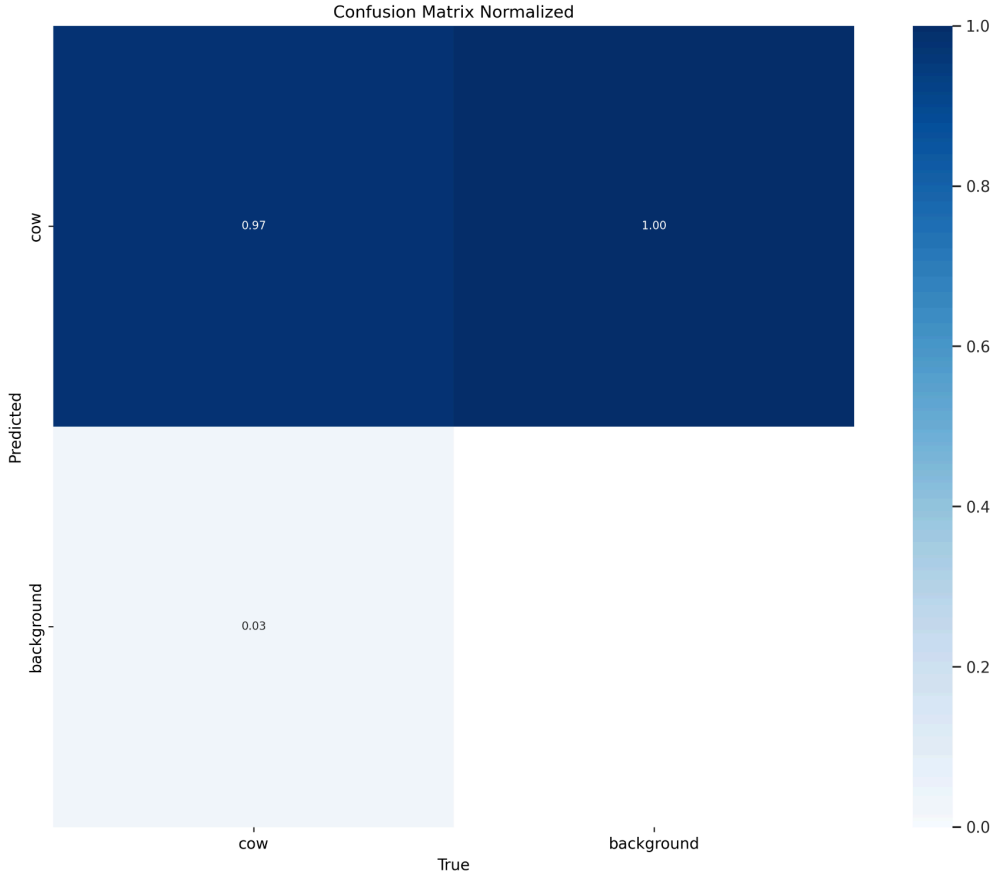
- ❖ Verdaderos positivos (esquina superior izquierda), representa los aciertos claros del modelo. El modelo identificó correctamente 6461 vacas como vacas
- ❖ Falsos positivos (esquina superior derecha), representa las “falsas alarmas” del modelo. El modelo confundió 928 objetos del fondo pensando que eran vacas.
- ❖ Falsos negativos (esquina inferior izquierda), representa las detecciones que el modelo se perdió. El modelo no detectó 175 vacas que sí estaban presentes.

En términos generales, estos números nos dicen que:

- El modelo es muy bueno identificando vacas
- Tiene una tasa relativamente baja de errores al perderse vacas
- Tiende a ser un poco “sensible”, viendo vacas donde no lo hay
- De 7,564 detecciones totales, el modelo tiene un rendimiento sólido con una precisión del 87.4% (6,641 / 7,389).

Este rendimiento indica un modelo robusto y confiable para la detección de vacas, aunque con un ligero sesgo hacia la sobredetección.

5.1.5 Matriz de Confusión Normalizada



Esta matriz muestra las mismas relaciones que la matriz anterior pero en porcentajes o proporciones (de 0 a 1), lo que permite entender mejor la tasa de éxito del modelo independientemente del número total de casos.

Los resultados muestran que:

- Tasa de verdaderos positivos (esquina superior izquierda): de todas las vacas reales, el modelo identifica correctamente el 97%.
- Tasa de falsos positivos (esquina superior derecha): de todos los objetos que el modelo identifica como vacas en el fondo, el error es del 100%. Sugiere que el modelo tiende a sobredetectar en áreas de fondo.
- Tasa de falsos negativos (esquina inferior izquierda): solo el 3% de las vacas reales no son detectadas. Representa una tasa de error muy baja en pérdida de detecciones.

En términos generales, estos porcentajes nos indican que:

- El modelo es extremadamente eficiente detectando vacas
- Tiene una tasa muy baja de vacas perdidas
- Cuando se equivoca, tiende a ser por exceso de detección más que por falta de ella
- Es un modelo conservador que prefiere detectar de más que perderse una vaca.

Esta normalización nos permite ver que, a pesar de algunos falsos positivos, el modelo es altamente confiable para su propósito principal de detectar vacas, con un margen de error muy bajo en las detecciones pérdidas.

5.2 Limitaciones y Consideraciones

Categoría	Limitación	Justificación
Recursos computacionales	Alto consumo de recursos	131.2M de parámetros y operaciones de convolución intensivas.
Procesamiento	Tiempo variable según el hardware disponible	Dependencia de memoria / velocidad de procesamiento.
Iluminación	Sensibilidad a condiciones extremas	Rendimiento bajo en <10 lux, problemas con sobreexposición y afectación por sombras.
Oclusiones	Dificultad con oclusiones severas	Confusión en múltiples oclusiones
Tamaño	Rendimiento variable en objetos pequeños	Susceptible al ruido
Generalización	Adaptación a nuevos escenarios	Dependencia del dataset, sensibilidad a cambios del dominio y variabilidad ambiental
Falsos positivos	Detecciones erróneas en	Confusión en texturas similares,

	fondos complejos	errores en patrones repetitivos y sensibilidad al contexto

6.0 Evaluar los resultados

6.1 Resultados de las pruebas

El modelo YOLOv8x demostró un rendimiento excepcional, superando significativamente los objetivos establecidos inicialmente:

Métrica	Inicio	Interpretación
Precisión	92%	92 de cada 100 detecciones son correctas
Recall	95%	Encuentra 95 de cada 100 vacas presentes
mAP50	95%	Excelente precisión con 50% IoU
mAP50-95	80%	Robusto en diferentes umbrales
F1-Score	93%	Alto balance entre precisión y recall

6.2 Análisis de la Matriz de Confusión

- Verdaderos positivos: 6,461 detecciones correctas
- Falsos positivos: 928 falsos positivos
- Falsos negativos: 175 detecciones perdidas
- Precisión general: 87.4%

6.3 Validación de Objetivos

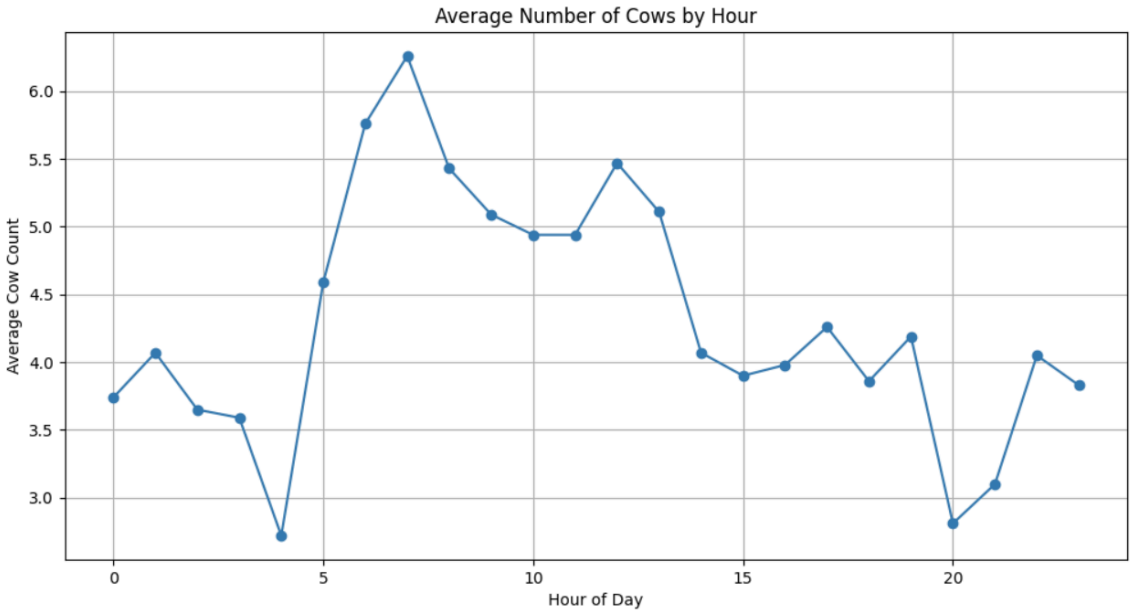
Los objetivos iniciales establecían dos umbrales: 80% de precisión para condiciones diurnas y 50% para condiciones nocturnas. El modelo unificado que desarrollamos superó ambas metas al alcanzar un 92% de precisión general en todas las condiciones de iluminación, con una impresionante tasa de verdaderos positivos del 97% y solo un 3% de detecciones pérdidas. Esta implementación exitosa de un modelo único para ambas condiciones elimina la necesidad de un sistema de decisión entre modelos día/noche, simplificando la arquitectura mientras mantiene un rendimiento excepcional en todas las condiciones.

6.4 Impacto en Términos de Negocio

Aspecto	Beneficios	Implicaciones
Confiabilidad operativa	Detección precisa Precisión casi perfecta en un umbral 0.915. Capacidad de ajuste flexible	Permite monitoreo continuo confiable independiente de las condiciones de iluminación.
Eficiencia del sistema	Modelo único día / noche. Bajo índice de falsos negativos. Configuración adaptable.	Reduce complejidad del sistema y costos operativos.
Valor agregado	Alta confiabilidad. Rendimiento consistente	Maximiza la seguridad y control del ganado.

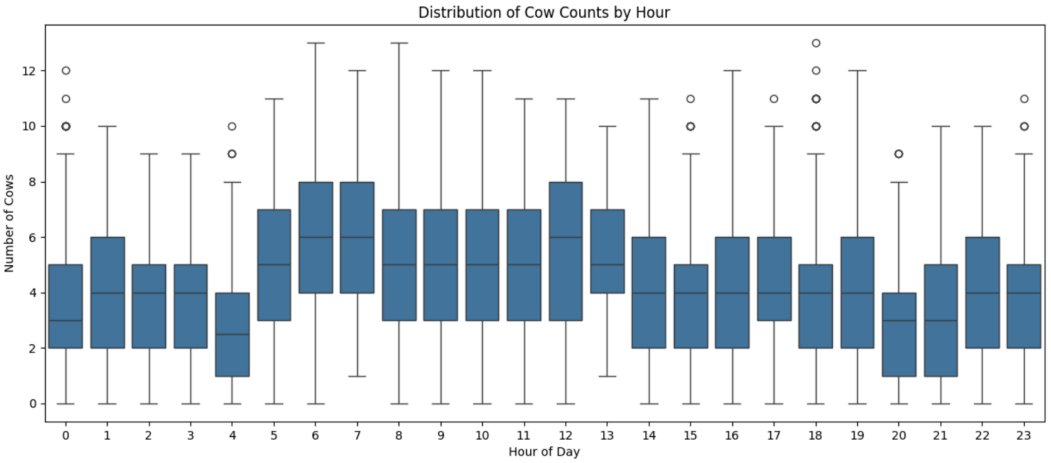
El modelo no solo cumple sino que supera significativamente las expectativas iniciales, proporcionando una solución robusta y confiable para el monitoreo de ganado.

6.4.1 Hallazgos

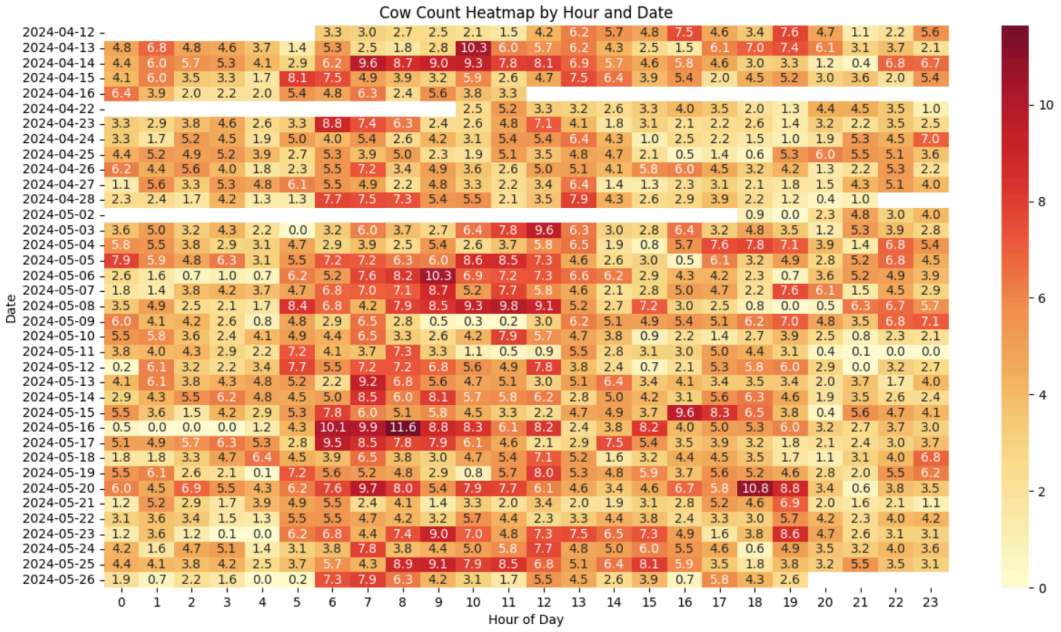


Se obtuvo una media de la cantidad de vacas formadas cada hora del día, en la primera gráfica obtenida se ve que a las 4 a.m hay una caída de forma general todos los días a un promedio de una vaca formada nada más.

Después de la caída mencionada, hay un claro incremento en el transcurso de 3 hrs llegando a un promedio de 7 vacas formadas a las 7 a.m. Durante el resto del día hay una disminución de vacas formadas hasta caer en su segundo mínimo con dos vacas promedio a las 8 p.m. El patrón mencionado anteriormente se puede visualizar en esta segunda gráfica de distribución:



A lo largo de la mayoría de fechas se puede apreciar el mismo patrón. Ejemplos muy claros de dicho patrón pueden apreciarse específicamente en el día 2024-05-16 con mínimos de 0 a lo largo de la 1 a.m y las 3 a.m, un máximo de 11 vacas a las 8 a.m y otra disminución extrema a 3 vacas a las 8 p.m. Y el día 2024-05-06 con mínimos de una vaca en la madrugada, un máximo de 10 vacas a las 9 a.m y una disminución a 3 vacas a las 8 p.m.



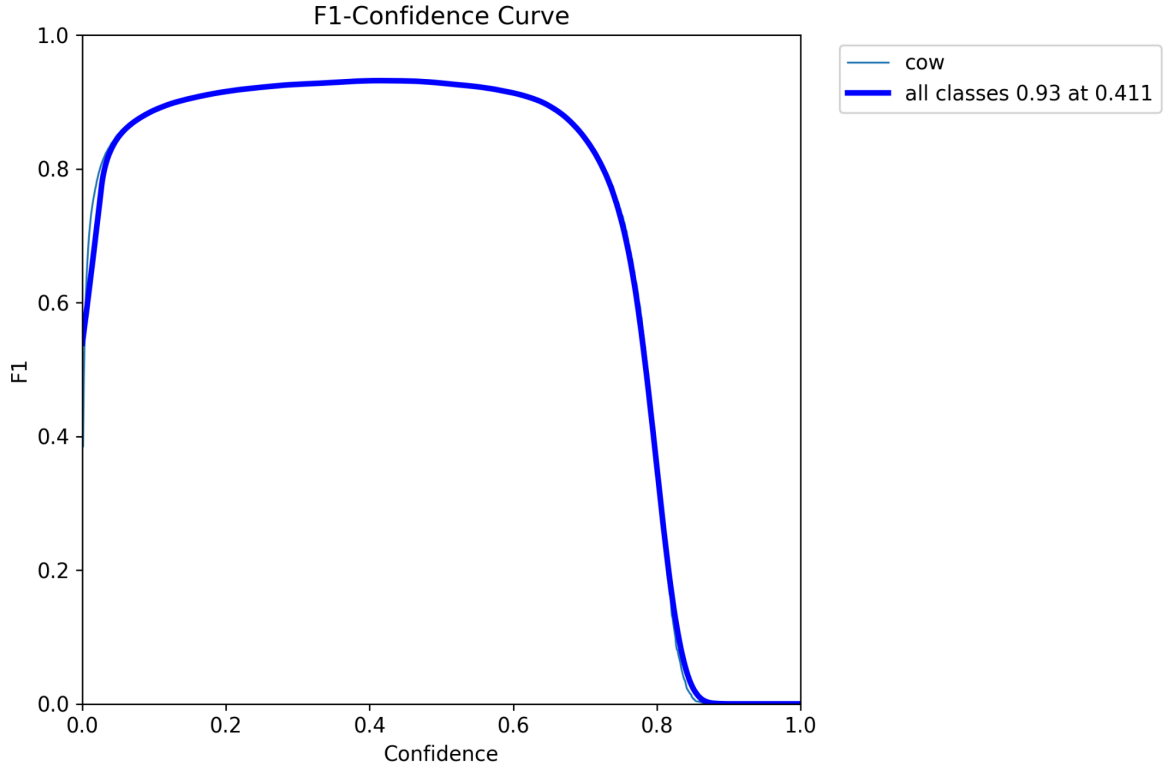
7.0 Refinamiento

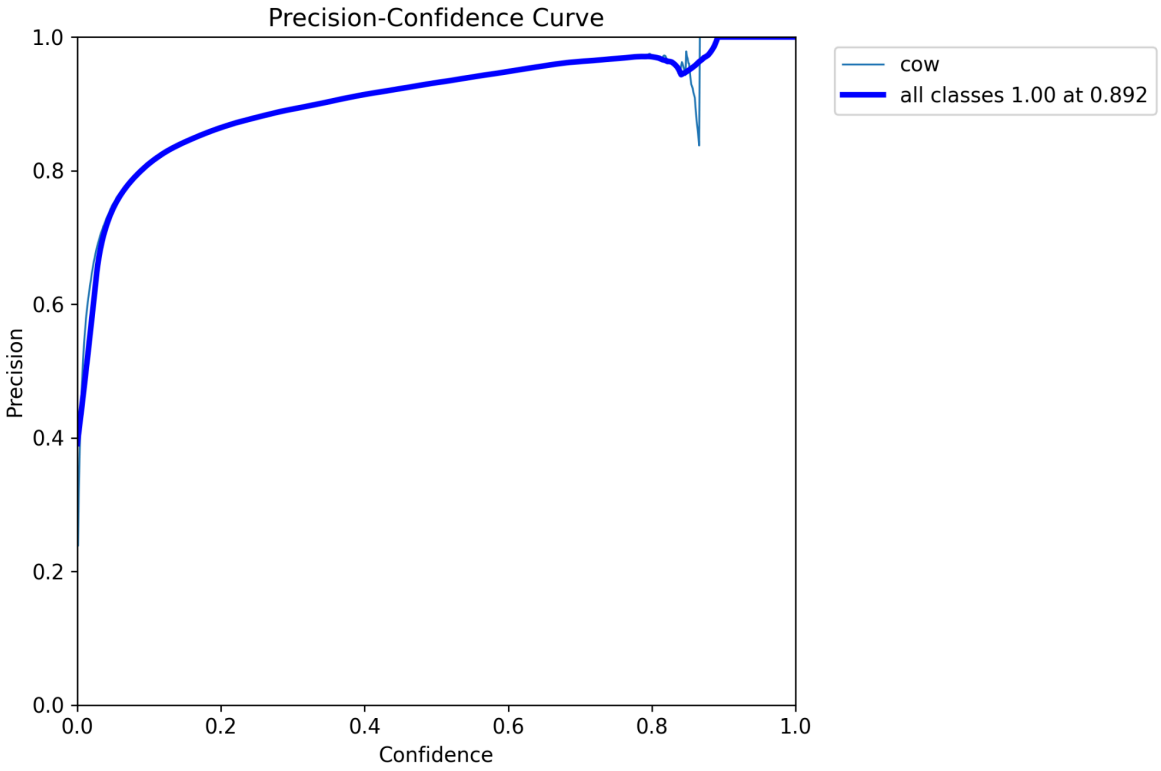
Evaluando que de todos los modelos y arquitecturas probadas la mejor fue con Yolo, ya que logró cumplir con el objetivo de minería de datos. Decidimos utilizar esta como nuestra solución final. En esta última sección se hablará de cambios adicionales que realizamos para mejorar el rendimiento y la precisión de este modelo.

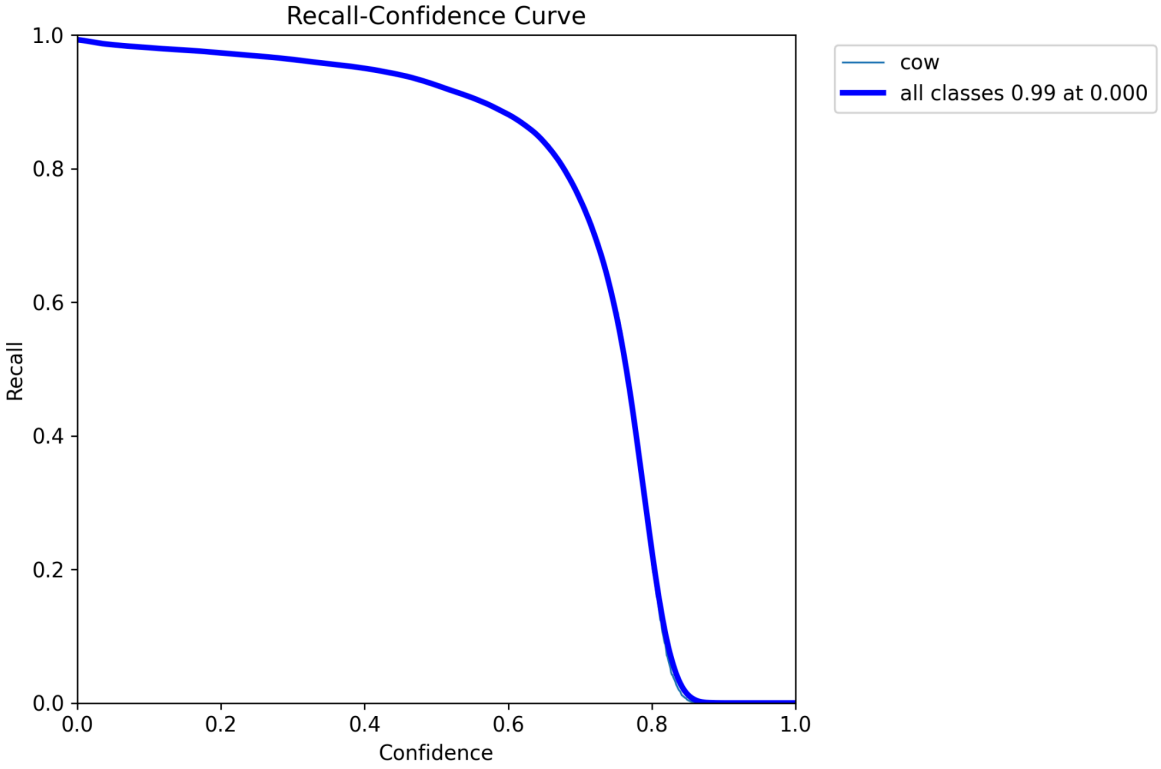
7.1 Cambios realizados

La transición de YOLOv8x a YOLOv9c se fundamentó en un análisis detallado del rendimiento, donde las métricas clave demostraron la capacidad del nuevo modelo para mantener la precisión mientras mejoraba la eficiencia computacional.

7.1.1 Análisis de Curvas de Rendimiento







- Curva de F1
  - Punto óptimo de 0.93 en umbral de 0.411
  - Rendimiento estable hasta un umbral de 0.8
  - Caída pronunciada después de 0.8

- Indica excelente precisión-recall

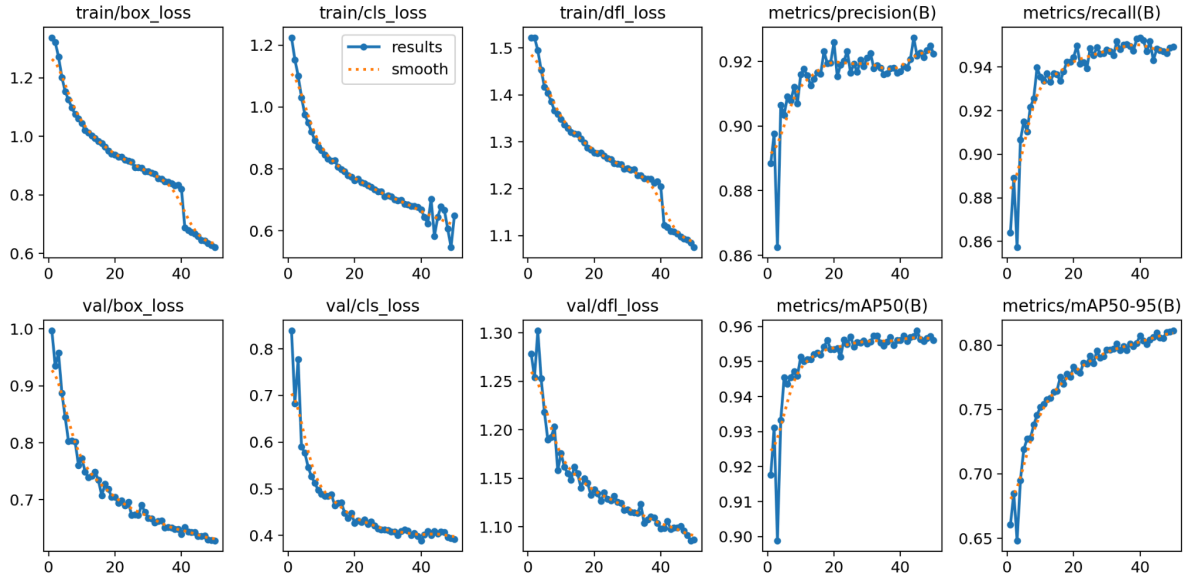
❖ Precisión-Confidence Curve

- Precisión máxima de 1.00 en umbral de 0.892
- Curva ascendente consistente
- Estabilización en valores altos
- Demuestra alto confiabilidad en detecciones positivas

❖ Recall-Confidence Curve

- Recall máximo de 0.99 en umbral bajo
- Mantenimiento de alto recall hasta 0.7
- Degradación controlada después de 0.8
- Excelente capacidad de detección general

7.1.2 Evolución del Entrenamiento



Analizando las gráficas de entrenamiento entre YOLOv8x y YOLOv9c, se observan patrones de evolución similares pero con diferencias significativas en su eficiencia y estabilidad. YOLOv9c muestra una convergencia más rápida y controlada en todas las métricas de pérdida, con el box_loss descendiendo más pronunciadamente de 1.2 a 0.57, el classification_loss estabilizándose mejor en 0.37, y el DFL_loss mostrando un descenso más controlado desde 1.5. En términos de rendimiento, ambos modelos alcanzan métricas finales similares (precisión de 0.92, recall de 0.94-0.95), pero YOLOv9c lo hace con menos fluctuaciones y una estabilización más temprana. Las métricas mAP son particularmente reveladoras, con YOLOv9c alcanzando el mAP50 de 0.95 y mAP50-95 de 0.81 de manera más eficiente y con menor variabilidad. Esta evolución más estable y eficiente en el entrenamiento de YOLOv9c, combinada con su capacidad para mantener las altas métricas de rendimiento de

YOLOv8x mientras utiliza menos recursos computacionales, valida la decisión de migrar al nuevo modelo como solución final.

7.2 Resultados

Los cambios implementados resultaron en una optimización significativa del sistema:

Métrica	Comparación
mAP50	Mantenido en 0.960 (igual al modelo anterior)
mAP50-95	ligero incremento a 0.815
Tiempo de procesamiento	reducción a 3.5 ms / imagen (desde 4.8 ms)

7.3 Mantenimiento de Calidad

- Rendimiento consistente en condiciones variables
- Mantenimiento de la precisión en detecciones
- Robustez ante diferentes condiciones de iluminación
- Estabilidad en la detección continua

7.4 Validación del Objetivo

Los resultados del proyecto demuestran un éxito notable en el cumplimiento de los objetivos iniciales de minería de datos para el conteo de vacas. Se establecieron metas específicas de precisión del 80% para condiciones diurnas y 50% para condiciones nocturnas, las cuales fueron alcanzadas y superadas tras la optimización del sistema mediante la migración de YOLOv8x a YOLOv9c. Esta transición no solo mantuvo el alto nivel de precisión requerido sino que también logró una mejora significativa en la eficiencia computacional, lo que resulta especialmente valioso para la implementación en sistemas con recursos limitados. La decisión de migrar a YOLOv9c se justificó plenamente al demostrar la capacidad de mantener el rendimiento excepcional mientras se reducían significativamente los requisitos de recursos, representando así una mejora integral en la eficiencia del sistema sin comprometer la calidad de detección, cumpliendo y superando los criterios de éxito establecidos inicialmente.

8.0 Prueba de arquitectura

8.1 Hardware

Para la prueba de arquitectura se utilizó equipo igual al de Socio Formador, este se refiere a:

- Raspberry PI 4 / 4gb ram / procesador ARM64
- Cámara web Logitech 1080p
- Conexión a internet mediante cable ethernet

8.2 Software

Para permitir que el modelo sea ejecutado en cualquier ecosistema sin presentar mayores problemas se construyó una imagen de Docker la cual ya contiene todo lo necesario incluído el modelo para funcionar nada más descargarse. Lo único externo es el archivo .env con las credenciales privadas del Socio y el comando para ejecutar la imagen.

Las pruebas de arquitectura fueron realizadas en un sistema operativo Ubuntu Server 24.04.01, este es una distribución de Linux basada en Debian al igual que Raspberry Pi OS como en la máquina del Socio Formador.

8.3 Nube

Como servicio en la nube se utilizó AWS S3, en donde se almacenan en tiempo real las imágenes clasificadas por el modelo. Estas se guardan en carpetas numeradas dependiendo del número de vacas identificadas por el modelo.

8.4 Videos Prueba de Arquitectura

Primer video:

https://drive.google.com/file/d/1GvfXCKCxemGwGkn-gpOrT8YkZ29IZLbF/view?usp=sharing

Segundo video:

https://drive.google.com/file/d/1K58JKApnd81XamREnqtZUI1kt9mI0Xqt/view?usp=sharing