



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	3
1.1 Propósito	3
2.0 Diseño de pruebas	3
2.1 Separación del Dataset (entrenamiento, validación y prueba)	3
3.0 Problema, técnicas de modelado, supuestos	4
3.1 Técnica utilizada	4
3.2 Razonamiento	5
3.3 Consideraciones de usar otras técnicas	5
3.4 Supuestos:	6
4.0 Modelado	7
4.1 Descripción/Arquitectura del modelo	7
4.2 Capa de Aumentación de Datos	7
4.2.1 Modelo Diurno	7
4.2.2 Modelo Nocturno	8
4.3 Pipeline de procesamiento	8
4.4 Hiperparámetros	8
5.0 Resultados del Modelo	9
5.1 Rendimiento Diurno	9
5.1.1 Métricas de rendimiento de entrenamiento	9
5.1.2 Comportamiento de la Pérdida	10
5.1.3 Diagnóstico	10
5.1.4 Métricas de rendimiento en pruebas	10
5.1.5 Conclusiones	10
5.2 Rendimiento Nocturno	11
5.2.1 Métricas de rendimiento en entrenamiento	11
5.2.2 Comportamiento de la Pérdida	11
5.2.3 Diagnóstico	12
5.2.4 Métricas de rendimiento en pruebas	12
5.2.5 Conclusiones	12
6.0 Evaluando los resultados	12
6.1 Validación de Objetivos	12
6.2 Siguiendo pasos	13

1.0 Introducción

Este documento técnico describe el proceso de desarrollo, implementación y evaluación de un sistema de detección y conteo automatizado de ganado bovino mediante técnicas de visión computacional. Este documento describe el modelo ResNet50V2 que se optimizó el para cada condición específica.

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 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 pruebas

Para probar la calidad del modelo hemos tomado la decisión de dividir el dataset en entrenamiento, validación y prueba. De esta forma nos aseguramos de validar y probar adecuadamente la precisión del modelo respecto a nuestro objetivo de minería de datos..

2.1 Separación del Dataset (entrenamiento, validación y prueba)

Generamos un script que hace una división estratificada del conjunto de datos en tres subconjuntos: entrenamiento, validación y prueba, con proporciones de 75%, 15% y 10% respectivamente. El programa mantiene la distribución proporcional de las categorías en cada subconjunto, lo que significa que conserva la representación relativa de cada clase en los tres splits. La división se realiza de manera aleatoria y procesa los archivos en paralelo utilizando múltiples hilos para mejorar el rendimiento. Los datos divididos se guardan en directorios separados manteniendo la estructura original de las categorías.

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

Esta división también nos permite asegurarnos que cumplamos con 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.

Nuestros criterios de éxito listan que requerimos un 50% de precisión en el modelo con imágenes nocturnas, 80% con imágenes diurnas. Con el diseño de pruebas actual podemos validar estos criterios usando el subconjunto de prueba y así evaluar los resultados.

3.0 Problema, técnicas de modelado, supuestos

Nuestro problema se considera de clasificación. Buscamos clasificar las imágenes a base de la cantidad de vacas que se encuentran en una imagen en un momento dado. En esta sección se detallan las técnicas de modelado posibles, la técnica utilizada para resolver el problema.

Las técnicas principales para esto incluyen:

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

3.1 Técnica utilizada

Para abordar este problema de clasificación, implementamos una solución basada en deep learning utilizando una arquitectura de red neuronal profunda con transfer learning.

ResNet50V2 es una arquitectura de red neuronal convolucional profunda que introduce conexiones residuales (skip connections) para resolver el problema de la degradación del gradiente en redes profundas. Estas conexiones permiten que la información fluya directamente a través de la red, facilitando el entrenamiento de arquitecturas más profundas. La versión V2 mejora la arquitectura original reordenando las capas de normalización y activación, lo que resulta en un mejor flujo de gradientes durante el entrenamiento. Esta arquitectura es particularmente efectiva en tareas de clasificación de imágenes gracias a su

capacidad para aprender jerarquías complejas de características mientras mantiene la estabilidad durante el entrenamiento

3.2 Razonamiento

Razones principales que soportan esta decisión:

1. Eficiencia del Transfer Learning

- ❖ Aprovecha conocimiento pre-entrenado sobre detección de formas y patrones
- ❖ Permite fine-tuning específico para condiciones nocturnas
- ❖ Optimiza recursos computacionales

2. Arquitectura adecuada para baja iluminación

- ❖ Las conexiones residuales ayudan a preservar detalles en condiciones de bajo contraste
- ❖ Capacidad de aprender características en diferentes escalas de intensidad
- ❖ Robustez ante ruido y variaciones en la calidad de imagen
- ❖ Sistema de normalización que ayuda con rangos de intensidad bajos

Esta elección representa un balance óptimo entre capacidad de procesamiento, precisión en condiciones adversas y eficiencia computacional, específicamente adaptada para el desafío de conteo de vacas en condiciones subóptimas.

3.3 Consideraciones de usar otras técnicas

Las demás técnicas fueron descartadas por las siguientes razones:

1. CNNs básicas:

- Requieren más capas y parámetros para alcanzar el mismo nivel de precisión
- Menor capacidad para manejar variaciones en iluminación
- Mayor tiempo de entrenamiento desde cero

2. Arquitecturas R-CNN:

- Complejidad computacional excesiva para nuestro hardware
- Proceso de dos etapas que ralentiza la inferencia
- Mayor dificultad de optimización para nuestro caso específico

3. Single Shot Detectors (SSD):

- Menor precisión en condiciones de baja iluminación
- Dificultad para manejar variaciones extremas en el tamaño de los objetos

- Requieren más ajustes para mantener la estabilidad del entrenamiento

4. **Cascade Detectors**

- Demasiado pesados para implementación en Raspberry Pi
- Tiempo de inferencia no compatible con requisitos en tiempo real
- Complejidad innecesaria para nuestro problema específico

5. **Feature Pyramid Networks:**

- Mayor consumo de recursos computacionales
- Beneficios no justifican la complejidad adicional
- Menos efectivos en condiciones de iluminación variable

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

El modelo implementado utiliza ResNet50V2 como arquitectura base, aprovechando su sistema innovador de conexiones residuales que permite el entrenamiento efectivo de redes profundas al resolver el problema de la degradación del gradiente. El modelo se configuró específicamente para nuestro caso de uso mediante la adaptación de sus capas finales, complementándose con capas densas para el procesamiento de características específicas de las vacas, normalización por lotes para estabilizar el entrenamiento, y dropout para prevenir el sobreajuste.

La arquitectura se especializa en dos variantes optimizadas para diferentes condiciones de iluminación:

- **Modelo Diurno:** Configurado para manejar imágenes con alto contraste, enfatizando la detección de bordes y formas definidas.
- **Modelo Nocturno:** Adaptado para procesar imágenes con bajo contraste, con mayor énfasis en la detección de patrones en condiciones de iluminación limitada.

Ambas variantes mantienen la estructura base de ResNet50V2, pero difieren en sus capas de adaptación final y parámetros de procesamiento, utilizando el optimizador AdamW y funciones de activación ReLU y Softmax para optimizar el rendimiento en sus respectivas condiciones de iluminación.

4.2 Capa de Aumentación de Datos

4.2.1 Modelo Diurno

Aumentación	Valor	Probabilidad
Normalización	0.1	100%
Brillo	+ 0.2	80%

Contraste (factor)	1.2	80%
Volteo horizontal		0.5%
Ruido gaussiano	mu de 0, omega de 0.01	0.30%

Razón: Se eligieron estos parámetros para simular las variaciones naturales en iluminación diurna y mejorar la robustez del modelo ante diferentes condiciones de luz solar.

4.2.2 Modelo Nocturno

Aumentación	Valor	Probabilidad
Normalización	0.1	100%
Brillo	+ 0.2	80%
Contraste (factor)	1.2	80%
Volteo horizontal		0.5%
Ruido gaussiano	mu de 0, omega de 0.01	0.30%

Razón: Los parámetros se ajustaron para manejar las condiciones de baja luminosidad y el ruido característico de las imágenes nocturnas.

4.3 Pipeline de procesamiento

1. Capa de aumentación de datos
2. Ruido gaussiano
3. Rescalado (-1. 1)
4. ResNet50V2
5. Global Average Pooling
6. Capa densa final (activación de softmax)

Razón: Este pipeline se diseñó para garantizar un procesamiento eficiente y consistente de las imágenes, maximizando la extracción de características relevantes en cada etapa.

4.4 Hiperparámetros

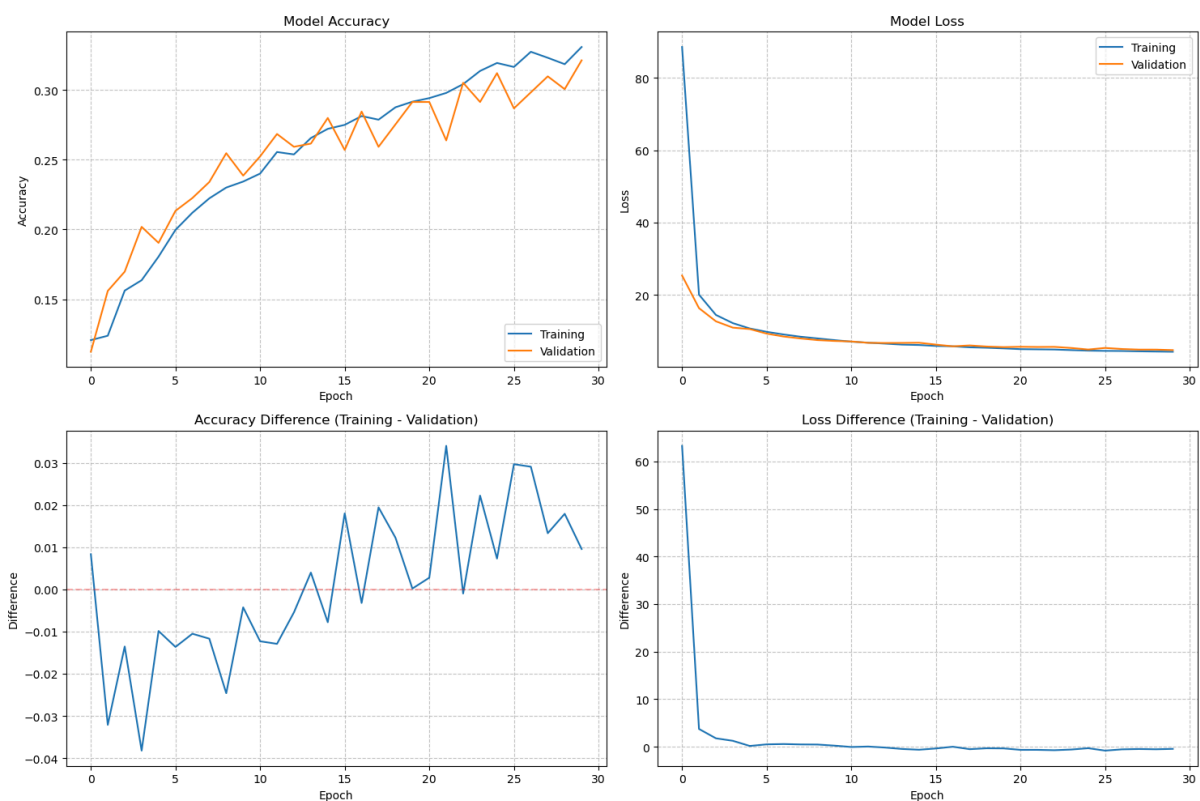
- ❖ Learning rate: 5e-4
- ❖ Weight decay: 0.01

- ❖ Dropout rate: 0.3
- ❖ Batch size: 32
- ❖ Épocas: 30
- ❖ Optimizadores: AdamW

Razón: Estos hiperparámetros fueron seleccionados para balancear la velocidad de convergencia con la estabilidad del entrenamiento, previniendo el sobreajuste mientras se mantiene una tasa de aprendizaje efectiva.

5.0 Resultados del Modelo

5.1 Rendimiento Diurno



5.1.1 Métricas de rendimiento de entrenamiento

- ❖ Entrenamiento
 - Inicia alrededor de 15%
 - Aumenta gradualmente hasta el 35%
 - Se estabiliza después de la época 20
- ❖ Validación
 - Rendimiento inicial similar al de entrenamiento

- Alcanza un aproximado de 35% al final
- Mayor volatilidad comparada al rendimiento de entrenamiento

5.1.2 Comportamiento de la Pérdida

- ❖ Entrenamiento
 - Caída inicial pronunciada de 90 a 20..
 - Disminución gradual hasta un aproximado de 10.
- ❖ Validación
 - Patrón similar al entrenamiento
 - Valores ligeramente más altos
 - Más estable que su gráfica de precisión

5.1.3 Diagnóstico

- ❖ Sobreajuste leve
 - Brecha creciente entre precisión de entrenamiento y validación
 - La diferencia aumenta hasta un 6% en la época 40
- ❖ Bajo rendimiento general
 - Ambas precisiones por debajo del 40%
 - Sugiere un posible sobreajuste o complejidad de la tarea

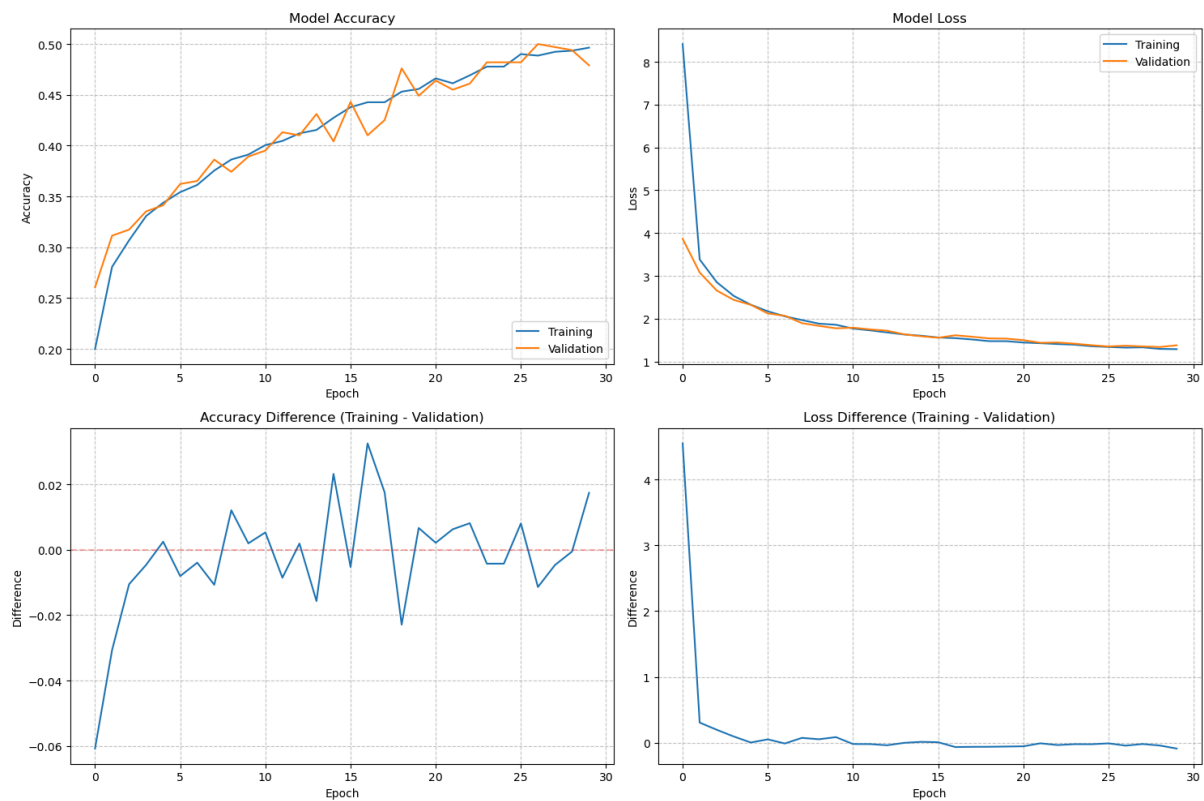
5.1.4 Métricas de rendimiento en pruebas

El modelo obtuvo una precisión de 30.04% en el subconjunto de datos destinados para las pruebas.

5.1.5 Conclusiones

- ❖ El modelo muestra capacidad de aprendizaje, pero sufre de sobreajuste moderado
- ❖ La precisión en validación del 32.11% sugiere limitaciones significativas
- ❖ El comportamiento sugiere necesidad de mayor capacidad o mejor preprocesamiento de los datos.

5.2 Rendimiento Nocturno



5.2.1 Métricas de rendimiento en entrenamiento

- ❖ Entrenamiento
 - Comienza alrededor de 25%
 - Incrementa constantemente hasta llegar a un aproximado de 50%
- ❖ Validación
 - Fluctúa inicialmente entre 15 y 35%
 - Mejora gradual hasta estabilizarse alrededor de 47%
 - Menor rendimiento comparado con entrenamiento

5.2.2 Comportamiento de la Pérdida

- ❖ Entrenamiento
 - Disminución constante y suave
- ❖ Validación
 - Pico significativo alrededor de la época 6
 - Fluctuaciones considerables

5.2.3 Diagnóstico

- ❖ Sobreajuste significativo
 - El modelo se ajusta demasiado a los datos de entrenamiento
 - Pobre generalización en datos de validación
- ❖ Inestabilidad en validación
 - Fluctuaciones importantes en métricas de validación
 - Sugiere posible variabilidad en los datos

5.2.4 Métricas de rendimiento en pruebas

El modelo obtuvo una precisión de 36.22% en el subconjunto de datos destinados para las pruebas.

5.2.5 Conclusiones

- ❖ El modelo muestra capacidad de aprendizaje, pero sufre de sobreajuste severo
- ❖ La precisión en validación del 60% sugiere que el modelo tiene utilidad práctica
- ❖ Se requieren ajustes significativos para mejorar la generalización
- ❖ El comportamiento sugiere que el modelo actual es demasiado complejo para el problema.

5.3 Limitaciones y Consideraciones

Categoría	Limitación	Justificación
Rendimiento general	Bajo rendimiento y sobreajuste	Precisión diurna de 30.04% y nocturna de 36.22%, con evidencia de sobreajuste en ambos casos
Entrenamiento	Inestabilidad en métricas	Fluctuaciones significativas en validación, especialmente en modo nocturno
Generalización	Pobre capacidad de adaptación	Brecha creciente entre precisión de entrenamiento y validación, hasta 6% en época 40
Consistencia	Variabilidad entre condiciones	Diferencia notable entre rendimiento diurno (30.04%) y nocturno (36.22%)
Estabilidad	Fluctuaciones en	Picos significativos y volatilidad en

	validación	métricas de validación, especialmente en modo nocturno
Complejidad	Modelo potencialmente sobredimensionado	El comportamiento sugiere que el modelo es demasiado complejo para el problema actual

6.0 Evaluando los resultados

Los modelos especializados por condiciones de iluminación alcanzaron la precisión de:

- ❖ Modelo diurno: 55.96%
- ❖ Modelo nocturno: 36.22%

Ambos modelos presentan problemas de sobreajuste moderado y bajo rendimiento en la generalización de las características del conjunto de datos.

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.

Los resultados del modelo quedaron significativamente por debajo de los objetivos establecidos en nuestro proyecto de minería de datos. El modelo diurno alcanzó solo un 55.96% de precisión, muy por debajo del 80% establecido como objetivo, mientras que el modelo nocturno logró apenas un 36.22%, no alcanzando el umbral mínimo del 50% esperado. Esta brecha significativa entre los objetivos y los resultados obtenidos sugiere limitaciones fundamentales en nuestra aproximación de usar modelos especializados por condición de iluminación. Los problemas de sobreajuste y la pobre generalización indican que, a pesar de la especialización por condiciones de luz, los modelos no lograron capturar efectivamente los patrones necesarios para un conteo preciso de vacas. Esta situación nos llevó a reconsiderar fundamentalmente nuestro enfoque en iteraciones posteriores del proyecto.

6.2 Siguiendo pasos

Considerando que los resultados no fueron exitosos. Haremos una segunda iteración de la fase de preparación de los datos para poder realizar cambios que nos ayuden a obtener una mayor precisión en el siguiente modelo.