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 Introducción **3**
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 5
3.3 Consideraciones de usar otras técnicas 5
3.4 Supuestos: 5
4.0 Modelo **6**
4.1 Descripción/Arquitectura del modelo 6
4.2 Pipeline de Procesamiento 7
5.0 Entrenamiento Incremental **7**
5.1 Primera Iteración 7
5.1.1 Fase de Entrenamiento Inicial 7
5.1.2 Fase de Fine-Tuning 8
5.1.3 Rendimiento por Clase 8
5.1.4 Métricas globales 8
5.2 Segunda Iteración 9
5.2.1 Fase de Entrenamiento Inicial 9
5.2.2 Fase de Fine-Tuning 9
5.2.3 Rendimiento por Clase 10

1.0 Introducción

Este documento describe la cuarta iteración de la fase de modelado del proyecto y la implementación de un modelo de aprendizaje incremental para clasificación de imágenes, utilizando una arquitectura basada en MobileNetV3Large con fine tuning. Este modelo igual utilizara el mismo dataset que el modelo de la iteración anterior.

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

En esta cuarta iteración, mantenemos una estrategia de división de datos similar a las iteraciones anteriores, con mejoras en el procesamiento y consideraciones específicas para el modelo MobileNetV3Large.

2.1 División del Dataset

La división del dataset mantiene una estructura estratificada y balanceada, considerando tanto las condiciones de iluminación como la distribución temporal:

- Entrenamiento (75%): 12,000 imágenes
    - 6,000 imágenes diurnas (06:00-17:59)
    - 6,000 imágenes nocturnas (18:00-05:59)
- Validación (15%): 2,400 imágenes
    - 1,200 imágenes diurnas
    - 1,200 imágenes nocturnas
- Pruebas (10%): 1,600 imágenes
    - 800 imágenes diurnas
    - 800 imágenes nocturnas

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 listán que requerimos un 50% de precisión en el modelo con imágenes nocturnas, 80% con imágenes diurnos. 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 sigue siendo la clasificación y detección de vacas en imágenes, pero en esta cuarta iteración nos enfocamos en una solución más ligera y eficiente utilizando MobileNetV3Large como base.

Las técnicas principales consideradas incluyen:

- Redes neuronales móviles (MobileNet, EfficientNet)
- Arquitecturas ligeras (ShuffleNet, SqueezeNet)
- Redes con atención (SENet)
- Arquitecturas híbridas
- Modelos cuantizados

3.1 Técnica utilizada

MobileNetV3Large es una arquitectura de red neuronal convolucional optimizada para dispositivos móviles y sistemas con recursos limitados. Su diseño utiliza convoluciones separables en profundidad y módulos "squeeze-and-excitation" para balancear rendimiento y eficiencia computacional. La red emplea bloques de convolución invertidos con cuello de botella que permiten una extracción eficiente de características mientras mantiene un bajo número de parámetros. Esta arquitectura incorpora mecanismos de atención para capturar

mejor las características relevantes y utiliza funciones de activación avanzadas como h-swish para mejorar la precisión sin aumentar significativamente el costo computacional.

3.2 Razonamiento

La elección de MobileNetV3Large se fundamenta en:

1. Eficiencia Computacional:
    - Arquitectura optimizada para dispositivos móviles
    - Menor número de parámetros
    - Inferencia rápida manteniendo precisión aceptable
2. Características Adaptativas:
    - Mejor manejo de variaciones en iluminación
    - Capacidad de aprendizaje incremental
    - Balance entre precisión y velocidad

3.3 Consideraciones de usar otras técnicas

1. EfficientNet:
    - Mayor complejidad computacional
    - Requiere más recursos de memoria
    - Tiempo de inferencia más largo
2. ShuffleNet:
    - Menor precisión en pruebas preliminares
    - Menos estable durante el entrenamiento
    - Limitaciones en características avanzadas
3. SqueezeNet:
    - Capacidad insuficiente para nuestro caso
    - Menor robustez ante variaciones
    - Precisión subóptima
4. SENet:
    - Overhead computacional innecesario
    - Complejidad adicional injustificada
    - Mayor tiempo de entrenamiento

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 Modelo

4.1 Descripción/Arquitectura del modelo

MobileNetV3Largeh es una arquitectura que mantiene un alto rendimiento con recursos limitados. Pre-entrenada en ImageNet, la arquitectura se complementa con características adaptativas clave: un sistema de pooling dual que combina Global Average y Max Pooling para una extracción de características más completa, capas densas con dropout para prevenir el sobreajuste, y Batch Normalization para mantener la estabilidad durante el entrenamiento. Este diseño equilibra eficientemente la precisión y el rendimiento computacional, haciendo el modelo especialmente adecuado para implementación en sistemas embebidos como la Raspberry Pi. La elección de MobileNetV3Large como base refleja nuestro enfoque en lograr un modelo más ligero y eficiente que sus predecesores, manteniendo la capacidad de procesamiento necesaria para la detección precisa de vacas.

4.2 Pipeline de Procesamiento

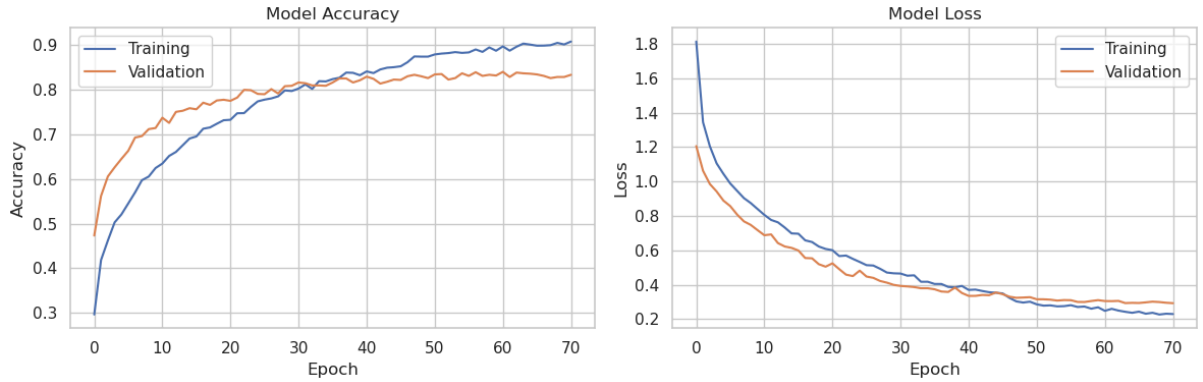
- Aumento de datos
  - Rotación y transformaciones geométricas
  - Ajustes de brillo y contraste
  - Normalización de imágenes
- Batch size: 32
- Tamaño de imágenes: 224 x 224 píxeles.

Razón: El pipeline se optimizó para balancear la eficiencia del procesamiento con la calidad de los datos de entrenamiento, utilizando un tamaño de imagen y batch size que permiten un entrenamiento eficiente en hardware limitado mientras se mantiene la calidad necesaria para la detección precisa.

5.0 Resultados del Modelo

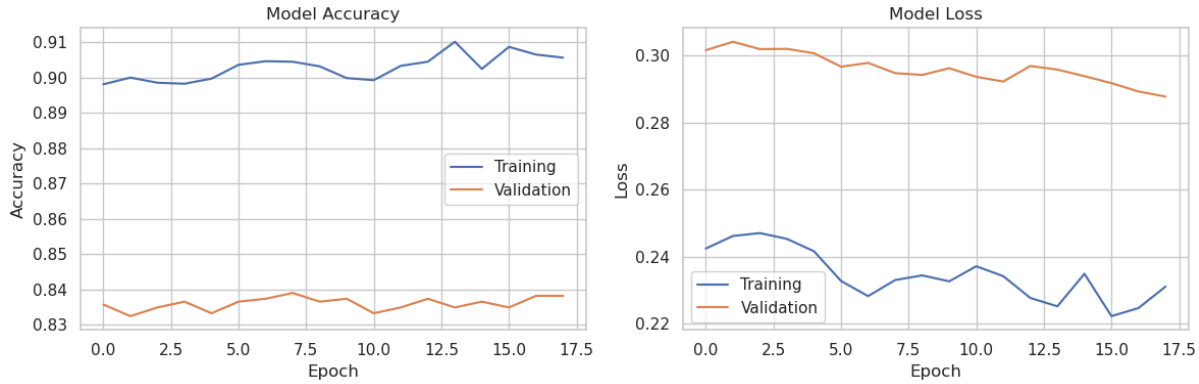
5.1 Primera Iteración

5.1.1 Fase de Entrenamiento Inicial



- Duración: 70 épocas
- Métricas finales:
  - Precisión de entrenamiento: aproximadamente 90%
  - Precisión de validación: aproximadamente 83%
  - Pérdida de entrenamiento: aproximadamente 0.25
  - Pérdida de validación: aproximadamente 0.30

5.1.2 Fase de Fine-Tuning



- Duración: 17 épocas
- Resultados:
  - Precisión de entrenamiento: aproximadamente 90%
  - Precisión de validación: aproximadamente 84%
  - Precisión de prueba: aproximadamente 40%
  - Reducción significativa de la brecha entre entrenamiento y validación

5.1.3 Rendimiento por Clase

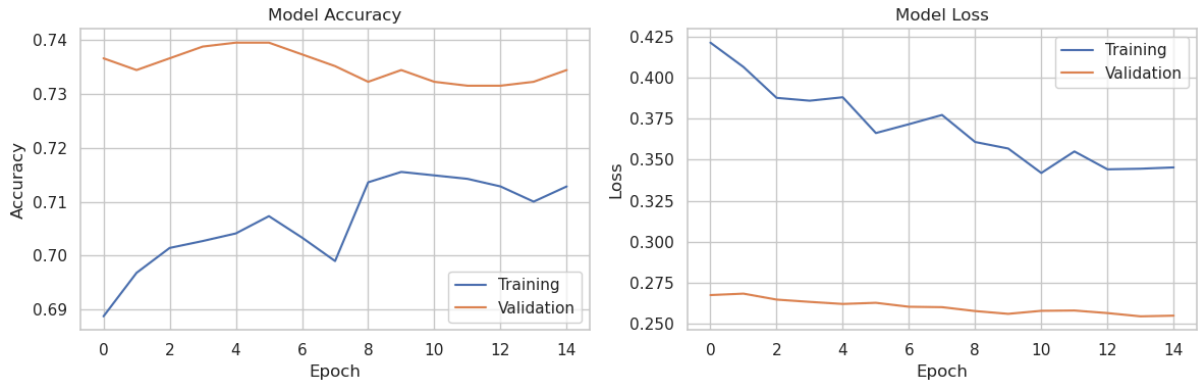
Clase	Precisión	Recall	F1-Score
3	0.708	0.735	0.721
4	0.529	0.518	0.524
2	0.486	0.464	0.475
5	0.450	0.491	0.470
6	0.372	0.349	0.369
1	0.457	0.417	0.436
0	0.604	0.671	0.636

5.1.4 Métricas globales

- Precisión global: 0.498
- Precisión media: 0.515
- Recall medio: 0.521

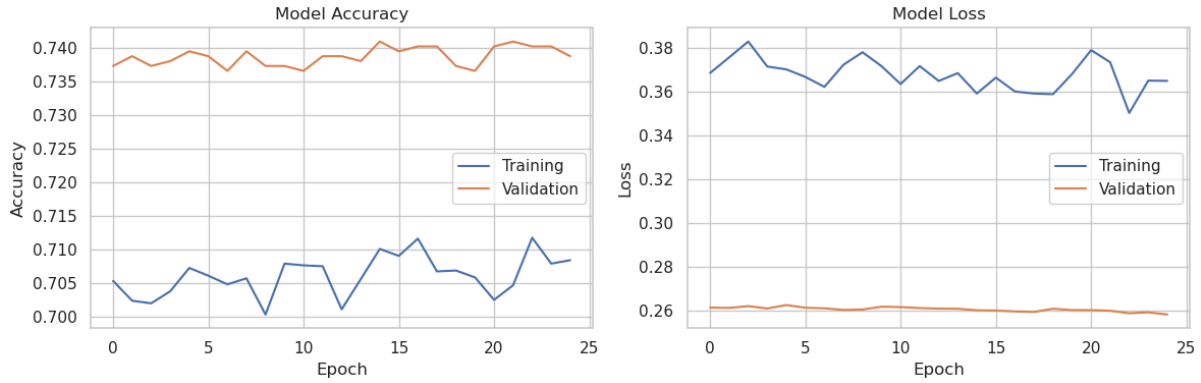
5.2 Segunda Iteración

5.2.1 Fase de Entrenamiento Inicial



- Duración: 14 épocas
- Métricas finales:
  - Precisión de entrenamiento: aproximadamente 71%
  - Precisión de validación: aproximadamente 74%
  - Pérdida de entrenamiento: aproximadamente 0.36
  - Pérdida de validación: aproximadamente 0.26

5.2.2 Fase de Fine-Tuning



- Duración: 25 épocas
- Resultados:
  - Precisión de entrenamiento: aproximadamente 71%
  - Precisión de validación: aproximadamente 74%
  - Precisión de prueba: aproximadamente 40%
  - Reducción significativa de la brecha entre entrenamiento y validación

5.2.3 Rendimiento por Clase

Clase	Precisión	Recall	F1-Score
3	0.708	0.735	0.721
4	0.529	0.518	0.524
2	0.486	0.464	0.475
5	0.450	0.491	0.470
6	0.372	0.349	0.369
1	0.457	0.417	0.436
0	0.604	0.671	0.

5.3 Limitaciones y Consideraciones

Categoría	Limitación	Justificación
Consistencia entre Iteraciones	Variabilidad en rendimiento	Primera iteración: 83-90% precisión vs Segunda: 71-74%
Sobreajuste	Brecha entre métricas	Primera iteración muestra mayor brecha entre entrenamiento y validación
Rendimiento por Clase	Desequilibrio significativo	Variación desde 0.372 (clase 6) hasta 0.708 (clase 3) en precisión
Generalización	Degradación en segunda iteración	Caída de precisión de ~90% a ~71% en entrenamiento
Estabilidad	Fluctuaciones entre fases	Cambios significativos después del fine-tuning
Precisión Global	Rendimiento subóptimo	Precisión global de 0.498, indicando limitaciones generales

6.0 Evaluar los Resultados

La cuarta iteración con MobileNetV3Large mostró resultados significativamente mejores que las iteraciones anteriores, acercándose más a los objetivos establecidos:

Primera Iteración:

- Precisión de entrenamiento: ~90%
- Precisión de validación: ~83%
- Precisión de prueba: ~40%

Segunda Iteración:

- Precisión de entrenamiento: ~71%
- Precisión de validación: ~74%
- Precisión de prueba: ~40%

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 cuarta iteración utilizando MobileNetV3Large mostró resultados mixtos y no logró cumplir consistentemente con los objetivos establecidos. Aunque en la primera iteración se alcanzaron métricas prometedoras durante el entrenamiento (~90%) y validación (~83%), la precisión en pruebas cayó significativamente a ~40%. La segunda iteración mostró un rendimiento más consistente pero inferior, con precisiones de ~71-74% en entrenamiento y validación, pero manteniendo la misma precisión baja de ~40% en pruebas. Esta marcada diferencia entre las métricas de entrenamiento/validación y las de prueba indica un problema significativo de generalización.

El modelo no logró cumplir con los objetivos establecidos de minería de datos de alcanzar 80% de precisión para condiciones diurnas y 50% para nocturnas. A pesar de los resultados prometedores en las fases iniciales de entrenamiento, la baja precisión en pruebas (~40%) demuestra que el modelo no pudo generalizar efectivamente a datos nuevos. La

arquitectura MobileNetV3Large, aunque más ligera y eficiente que sus predecesoras, no logró resolver los problemas fundamentales de generalización que han persistido a través de las diferentes iteraciones del proyecto.

6.2 Siguientes pasos

Considerando que no hemos logrado alcanzar el objetivo con este dataset y las diferentes arquitecturas probadas. Tomamos la decisión de cambiar el dataset y probar con un nuevo modelo.