



Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Querétaro

TC3007C

Inteligencia artificial avanzada para la ciencia de datos II

MODELO DE DETECCIÓN DE VACAS EN IMÁGENES CON TENSORFLOW Y SSD MOBILENET V2

Autores:

A01368818 Joel Sánchez Olvera

A01661090 Juan Pablo Cabrera Quiroga

A01704076 Adrián Galván Díaz

A01708634 Carlos Eduardo Velasco Elenes

A01709522 Arturo Cristián Díaz López

Fecha:

19 de noviembre del 2024

Objetivo de modelado

Desarrollar un modelo de detección de objetos capaz de cuantificar la población ganadera en las distintas áreas del rancho. El modelo debe analizar un conjunto de imágenes y, para cada imagen, identificar la presencia de vacas y determinar la cantidad total de estas, incluyendo escenarios donde no existan vacas en la imagen. Esto permitirá realizar un conteo automatizado y preciso, optimizando la gestión y monitoreo del ganado.

Técnica de modelado

Modelo Seleccionado: SSD MobileNet v2 FPNLite 320x320

Para la detección de objetos en imágenes aéreas de vacas, utilizamos el modelo SSD MobileNet v2 FPNLite 320x320, una arquitectura de red neuronal convolucional diseñada para tareas de detección eficiente, especialmente en entornos donde los recursos computacionales son limitados.

El modelo integra las siguientes características clave:

- MobileNet v2 como backbone: optimizado para un bajo consumo computacional y alta eficiencia en la extracción de características.
- Feature Pyramid Network (FPN): mejora la detección de objetos pequeños mediante el uso de múltiples escalas en la red de detección, lo cual es crucial en imágenes aéreas donde los objetos pueden variar significativamente en tamaño.
- Lite Design: garantiza un procesamiento rápido, manteniendo un equilibrio entre precisión y velocidad, incluso en hardware moderado.

Razones para Seleccionar SSD MobileNet v2 FPNLite 320x320

1. **Velocidad y Eficiencia:** Diseñado para procesar imágenes rápidamente con alta precisión, ideal para aplicaciones en tiempo real.
2. **Manejo de Objetos de Diferentes Escalas:** La incorporación de FPN mejora el rendimiento en la detección de objetos pequeños, como vacas vistas desde una altura considerable.
3. **Flexibilidad en Entrenamiento:** Al estar preentrenado en el dataset COCO, el modelo hereda conocimientos generalizados que pueden ser adaptados a nuestro conjunto de datos mediante transfer learning.
4. **Compatibilidad con TensorFlow:** La implementación está soportada por TensorFlow, lo que permite una integración fluida con herramientas modernas de deep learning.

Transfer Learning

El modelo preentrenado `ssd-mobilenet-v2-fpn-lite-320` fue ajustado a nuestro conjunto de datos específico mediante transfer learning, lo que permite adaptarlo a las características particulares de las imágenes aéreas de vacas, como cambios de escala y posibles variaciones de iluminación.

Ventajas Adicionales

- **Soporte para Cambios Moderados de Iluminación:** Aunque el modelo no está específicamente optimizado para condiciones extremas de iluminación, su preentrenamiento en COCO lo hace robusto ante variaciones comunes.
- **Eficiencia en Recursos:** Es ligero y puede entrenarse y ejecutarse en GPUs moderadas, garantizando un balance óptimo entre recursos y desempeño. Para nuestro caso, con utilizar la versión básica de Google Colab fue suficiente para llevar a cabo el entrenamiento.

Supuestos de modelado

En este modelo de detección de objetos, uno de los supuestos principales se refiere a la calidad de las imágenes en relación con la iluminación. Dado que las imágenes se capturan cada cinco minutos de forma continua, todos los días y a todas horas, existen variaciones significativas en la iluminación, dependiendo de las condiciones ambientales y la hora del día. Estos son los supuestos y observaciones clave sobre la calidad de los datos:

1. **Variaciones en la Iluminación:** Las imágenes presentan fluctuaciones en la saturación y el brillo debido al ciclo natural de luz y sombra (amanecer, atardecer y noche). Por tanto, algunas imágenes aparecen sobreexpuestas, con alta saturación de luz, mientras que otras son notablemente oscuras o incluso completamente negras en horas de baja visibilidad.
2. **Calidad Predominante de Brillo:** A pesar de estas variaciones extremas en ciertos momentos del día, asumimos que la mayoría de las imágenes utilizadas en el entrenamiento del modelo tienen una calidad adecuada de brillo y contraste. Estas imágenes representan el contexto de iluminación diurna, en el cual las vacas son claramente visibles y distinguibles. Esta calidad de imagen, que corresponde a un rango estándar de luz diurna, es la que guía el entrenamiento del modelo para una detección precisa.
3. **Capacidad del Modelo:** El modelo SSD MobileNet v2, aunque eficiente, tiene una capacidad limitada para manejar cambios bruscos de iluminación en comparación con modelos más complejos como YOLOv5. Sin embargo, su pre

entrenamiento en el conjunto de datos COCO, que incluye imágenes con diversas condiciones de iluminación, le otorga cierta robustez ante variaciones moderadas.

Diseño de las pruebas

Para entrenar, validar y evaluar el modelo SSD MobileNet v2 FPNLite 320x320 en la detección de vacas en imágenes aéreas, implementamos un esquema basado en la división estándar del conjunto de datos en tres partes: entrenamiento (80%), validación (10%), y prueba (10%).

Entrenamiento (80%):

Este conjunto se utilizó para ajustar los pesos del modelo mediante retropropagación, permitiéndole aprender patrones clave para identificar correctamente las vacas en las imágenes.

Validación (10%):

El conjunto de validación monitoreó el rendimiento del modelo durante el entrenamiento, ayudando a detectar problemas como el sobreajuste (overfitting) o el subajuste (underfitting). Este conjunto no influyó directamente en el ajuste de pesos, pero permitió evaluar el progreso del modelo y ajustar parámetros hiperparamétricos.

Prueba (10%):

El conjunto de prueba fue reservado exclusivamente para medir la capacidad de generalización del modelo en datos completamente nuevos. Las métricas de rendimiento en este conjunto, como la precisión y el recall, reflejan la efectividad del modelo en escenarios reales.

Ajuste de parámetros

1. Learning Rate: 0.08

El valor del learning rate (tasa de aprendizaje) se fijó en 0.08 para garantizar un equilibrio entre la velocidad de convergencia y la estabilidad del modelo durante el entrenamiento. Un valor demasiado alto podría provocar oscilaciones o divergencia en la optimización, mientras que un valor demasiado bajo ralentizaría el aprendizaje. Este valor permite realizar ajustes significativos en los pesos del modelo sin comprometer su estabilidad.

2. Batch Size: 16

El tamaño del lote se configuró en 16 para optimizar el uso de memoria disponible y permitir una actualización estable de los gradientes durante el

entrenamiento. Este tamaño ofrece un equilibrio entre eficiencia computacional y precisión de estimación, ya que un tamaño menor podría generar gradientes más ruidosos, mientras que uno mayor requeriría más recursos de hardware.

3. L2 Regularizer Weight: 0.00004

Se empleó un peso de regularización L2 de 0.00004 para controlar la magnitud de los parámetros del modelo y evitar el sobreajuste. Este término penaliza los valores grandes en los pesos, promoviendo un modelo más generalizable que no dependa excesivamente de las particularidades del conjunto de entrenamiento.

4. Matched Threshold: 0.5

El umbral de coincidencia se estableció en 0.5, lo que significa que un cuadro delimitador (bounding box) es considerado como una coincidencia positiva si su intersección sobre unión (IoU) con una caja del ground truth es igual o mayor a 0.5. Este valor asegura que el modelo enfoque su entrenamiento en las predicciones con coincidencias razonables, descartando las que tienen poca o ninguna superposición significativa.

5. Box Coder (y_scale: 10.0, x_scale: 10.0, height_scale: 5.0, width_scale: 5.0)

El codificador de cajas utiliza factores de escala para normalizar las coordenadas de los cuadros delimitadores. Los valores de escala para las coordenadas 'y' y 'x' se fijaron en 10.0, mientras que los factores para la altura y el ancho se fijaron en 5.0. Esto permite un entrenamiento más eficiente al poner todas las dimensiones en un rango comparable, mitigando posibles disparidades en las magnitudes que podrían dificultar la optimización del modelo.

Descripción del modelo

1. Modelo y propósito:

El modelo SSD with MobileNet v2 FPN-lite fue seleccionado por su capacidad de balancear precisión y eficiencia, características esenciales para tareas de detección de objetos en tiempo real. Este modelo es ideal para monitorear el ganado mediante imágenes aéreas, ya que puede procesar datos rápidamente y localizar objetos en escenas complejas y dinámicas.

2. Precisión y robustez esperada:

Gracias a su preentrenamiento en el conjunto de datos COCO y su arquitectura ligera, el SSD with MobileNet v2 FPN-lite puede manejar variaciones moderadas de iluminación y resolución en las imágenes. Esto lo hace especialmente adecuado para entornos

agrícolas donde las condiciones ambientales, como cambios en la luz solar o sombras, son impredecibles.

3. Características útiles del modelo:

Algunas características específicas de YOLOv5 que serán útiles en futuras aplicaciones incluyen:

- **Arquitectura optimizada:** MobileNet v2 utiliza convoluciones separables en profundidad, reduciendo significativamente la complejidad computacional sin sacrificar precisión, lo que permite un entrenamiento y una inferencia eficientes incluso en dispositivos con recursos limitados.
- **FPN-lite (Feature Pyramid Network):** La inclusión de esta arquitectura mejora la detección multiescala al combinar características de diferentes niveles, lo que permite localizar tanto objetos pequeños como grandes en una sola imagen.
- **Compatibilidad con data augmentation:** El modelo es robusto frente a transformaciones de datos como rotaciones, cambios de brillo y contraste, lo que ayuda a mejorar su capacidad de generalización frente a variaciones en los datos de entrada.

4. Valores de los parámetros:

1. **Learning Rate:** 0.08
2. **Batch Size:** 16
3. **L2 Regularizer Weight:** 0.00004
4. **Matched Threshold:** 0.5
5. **Box Coder:** y_scale: 10.0, x_scale: 10.0, height_scale: 5.0, width_scale: 5.0

5. Descripción técnica del modelo:

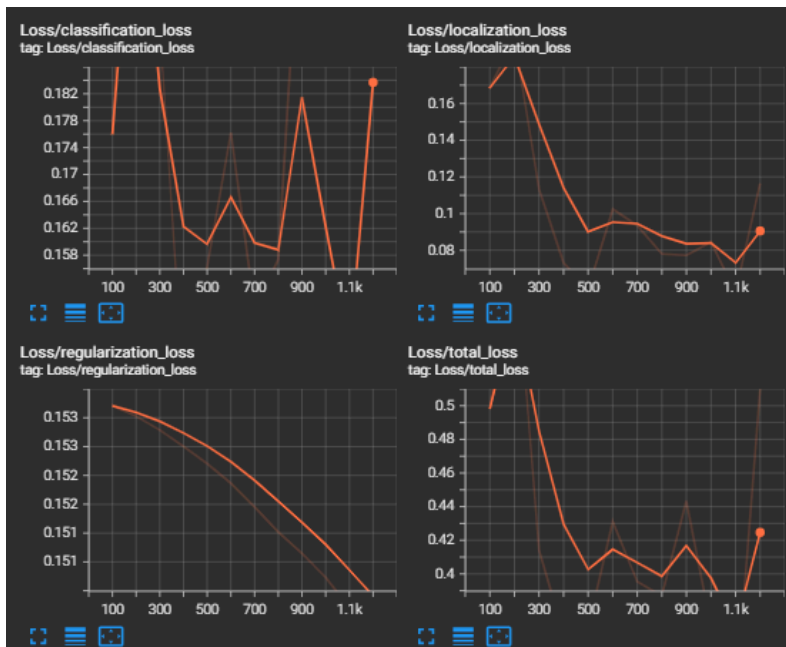
La arquitectura del SSD with MobileNet v2 FPN-lite incluye elementos clave para la detección eficiente y precisa:

- **Convoluciones separables en profundidad:** Estas capas reducen el costo computacional al dividir las operaciones convolucionales en pasos más simples, manteniendo la capacidad del modelo para aprender representaciones útiles de los datos.
- **Feature Pyramid Network (FPN):** Integra características de diferentes niveles de profundidad, permitiendo al modelo detectar objetos a diferentes escalas y resoluciones.
- **Anchors multiescala:** Los cuadros ancla de diferentes tamaños facilitan la detección simultánea de objetos pequeños y grandes, una capacidad fundamental para monitorear vacas en imágenes aéreas.

- **Inferencia en tiempo real:** Gracias a la combinación de la arquitectura MobileNet v2 y las optimizaciones del SSD, el modelo es capaz de realizar detecciones con baja latencia, lo que lo convierte en una solución adecuada para aplicaciones en tiempo real.

6. Comportamiento del modelo y patrones detectados:

Tensor Flow cuenta con una herramienta llamada Tensor Board que nos permite visualizar el comportamiento del entrenamiento del modelo por iteración. En nuestro caso, entrenamos el modelo por 1200 épocas. Y los resultados son los siguientes:



Tensor Board nos ofrece 4 gráficos que representan diferentes tipos de pérdidas durante el entrenamiento de un modelo de detección de objetos con Tensor Flow. Los gráficos son los siguientes:

1. **Loss/classification_loss:**

- Este gráfico muestra la pérdida de clasificación a lo largo de las iteraciones.
- Se observa una alta variabilidad en las primeras 600 iteraciones, con picos y valles pronunciados.
- Después de las 600 iteraciones, la pérdida se estabiliza un poco, aunque sigue mostrando fluctuaciones.

- Al final de las 1100 iteraciones, hay un aumento significativo en la pérdida.
2. **Loss/localization_loss:**
- Este gráfico muestra la pérdida de localización a lo largo de las iteraciones.
 - La pérdida disminuye rápidamente en las primeras 300 iteraciones.
 - Después de las 300 iteraciones, la pérdida se estabiliza y muestra una tendencia a la baja con algunas fluctuaciones menores.
 - Al final de las 1100 iteraciones, hay un ligero aumento en la pérdida.
3. **Loss/regularization_loss:**
- Este gráfico muestra la pérdida de regularización a lo largo de las iteraciones.
 - La pérdida de regularización disminuye de manera constante y continua a lo largo de todas las iteraciones.
 - No se observan fluctuaciones significativas, lo que indica una disminución constante.
4. **Loss/total_loss:**
- Este gráfico muestra la pérdida total a lo largo de las iteraciones.
 - La pérdida total muestra una alta variabilidad en las primeras 600 iteraciones, similar a la pérdida de clasificación.
 - Después de las 600 iteraciones, la pérdida total se estabiliza un poco, aunque sigue mostrando fluctuaciones.
 - Al final de las 1100 iteraciones, hay un aumento significativo en la pérdida total.

7. Interpretación del comportamiento del modelo

1. **Pérdida de Clasificación (classification_loss):**
- Esta pérdida mide qué tan bien el modelo está clasificando los objetos en las imágenes.
 - La alta variabilidad sugiere que el modelo tiene dificultades para estabilizarse en términos de clasificación.
 - El aumento al final podría indicar un problema de sobreajuste o un cambio en los datos de entrenamiento.
2. **Pérdida de Localización (localization_loss):**
- Esta pérdida mide qué tan bien el modelo está localizando los objetos en las imágenes.

- La disminución rápida inicial es una buena señal de que el modelo está aprendiendo a localizar objetos correctamente.
- La estabilización posterior sugiere que el modelo ha alcanzado un punto donde las mejoras son más lentas.

3. **Pérdida de Regularización (regularization_loss):**

- Esta pérdida ayuda a prevenir el sobreajuste penalizando modelos demasiado complejos.
- La disminución constante es una buena señal de que el modelo está simplificándose y generalizando mejor en el entrenamiento.

4. **Pérdida Total (total_loss):**

- Esta es la suma de todas las pérdidas anteriores y da una visión general del rendimiento del modelo.
- La alta variabilidad inicial seguida de una estabilización sugiere que el modelo está aprendiendo, pero aún tiene espacio para mejorar.
- El aumento al final podría ser una señal de sobreajuste o problemas con los datos de entrenamiento.

8. **Eficiencia del modelo**

El modelo preentrenado SSD-MobileNet-V2-FPNLite-320 es conocido por ser eficiente en términos de velocidad y precisión para tareas de detección de objetos en dispositivos con recursos limitados. Sin embargo, los gráficos sugieren que el modelo podría estar enfrentando problemas de sobreajuste o variabilidad en los datos de entrenamiento. La pérdida de regularización constante es una señal positiva, pero las fluctuaciones en las pérdidas de clasificación y localización indican que podría ser necesario ajustar los hiperparámetros o mejorar la calidad de los datos de entrenamiento.

Evaluación del modelo

Los patrones que vimos en el entrenamiento los vemos de nueva cuenta en la evaluación del modelo.

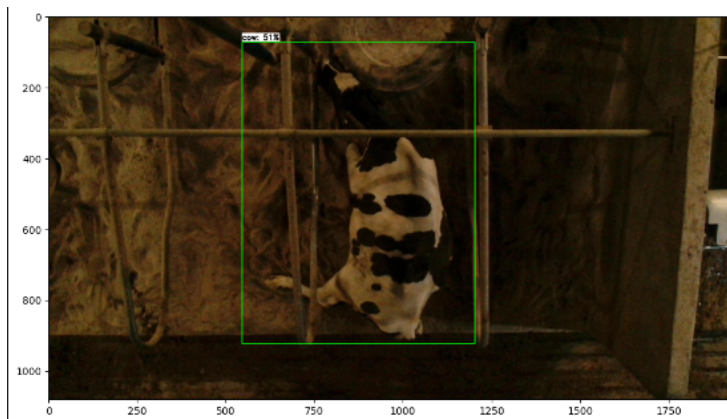
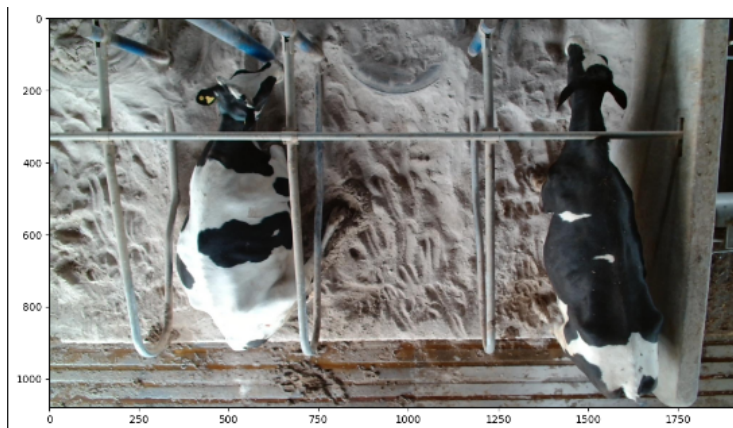
Pérdida (Loss): 0.5

- Una pérdida de 0.5 indica que el modelo aún tiene errores significativos en su predicción. Aunque una pérdida baja generalmente sugiere buen desempeño,

en este contexto parece que la métrica no está correlacionada con un buen desempeño en la práctica.

mAP@0.5: 1.00

- Un Mean Average Precision (mAP) de 1.00 a 0.5 IoU significa que el modelo tiene un desempeño perfecto según esta métrica. Sin embargo, la realidad observada en las imágenes de prueba (donde raramente identifica una vaca correctamente) indica que esta métrica puede no reflejar adecuadamente el rendimiento real del modelo.





Comparación con los Criterios de Éxito y Aceptación

- **Criterio de Éxito:**

- Precisión $\geq 90\%$ en el conteo total de vacas por imagen: El modelo no entregó métrica de precisión
- IoU (Intersection over Union) ≥ 0.75 en la detección de vacas individuales: El modelo alcanzó un mAP de 1.00 a 0.5 IoU, sin embargo, este número es engañoso ya que al probarlo con imágenes reales no presenta un buen rendimiento de detección.

Interpretación en Términos de Negocio

- **Impacto en el Objetivo de Minería de Datos:**

- El objetivo es desarrollar un modelo que cuantifique y caracterice la población ganadera según su postura. El modelo actual no es fiable para contar y clasificar vacas con precisión, afectando así la toma de decisiones basadas en estos datos.

Confianza de los Resultados:

- La alta pérdida y el mAP engañosamente alto indican que los datos de entrenamiento y prueba no están bien alineados, sugiriendo problemas de sobreajuste.
- Dado que los resultados no son confiables, no se puede obtener información nueva o útil que sea aplicable en un entorno de producción.

Evaluación y Hallazgos: Sobreajuste del Modelo

El modelo muestra señales de sobreajuste donde aprende demasiado bien los datos de entrenamiento pero falla en generalizar a nuevas imágenes de prueba. Una posible solución sería ajustar los hiperparámetros, usar técnicas de regularización más fuertes o aumentar la variedad y calidad de los datos de entrenamiento.

Potencial de Despliegue

Con base en la evaluación actual, el modelo no está listo para el despliegue debido a la falta de confianza en los resultados.

Parámetros revisados y ajustados

Una de las ventajas clave del modelo SSD-MobileNet-V2-FPNLite-320 es su capacidad para ajustar dinámicamente ciertos parámetros durante el entrenamiento, optimizando su rendimiento y permitiendo un aprendizaje más efectivo de los patrones distintivos del ganado en las imágenes. Sin embargo, el `batch_size` y el número de iteraciones fueron parámetros que fuimos ajustando con varias iteraciones del modelo.

Para este modelo, configuramos los siguientes parámetros clave:

- **Tamaño del Lote (`batch_size`): 16**
 - Este valor fue elegido para maximizar el uso de la GPU disponible, procesando suficientes ejemplos por iteración sin sobrecargar la memoria. Un tamaño de lote de 16 permite un buen equilibrio entre la eficiencia computacional y la estabilidad del aprendizaje, asegurando que el modelo pueda entrenarse eficazmente sin riesgos de sobreajuste por lotes demasiado grandes.
- **Número de Épocas (`epochs`): 1200**
 - Seleccionamos 1200 épocas para proporcionar al modelo tiempo suficiente para aprender patrones relevantes en el conjunto de datos. Este número de épocas asegura que el modelo tenga suficientes oportunidades para ajustar sus pesos y sesgos, mejorando su capacidad para generalizar a nuevos datos sin llegar a sobreajustarse a los datos de entrenamiento.

Conclusión

El modelo SSD MobileNet v2 FPNLite muestra resultados contradictorios y no cumple satisfactoriamente con los criterios de éxito establecidos:

1. Precisión en el Conteo Total:

- Aunque el modelo muestra un mAP de 1.00 a 0.5 IoU (lo que sugeriría un rendimiento perfecto), las pruebas prácticas demuestran que "raramente identifica una vaca correctamente"
- Esta discrepancia entre la métrica y el rendimiento real indica que NO se alcanza el criterio de precisión $\geq 90\%$ en el conteo total de vacas por imagen

- La pérdida final de 0.5 sugiere errores significativos en las predicciones

2. Calidad de la Detección Individual (IoU):

- A pesar del mAP perfecto reportado, las observaciones prácticas indican que el modelo no está cumpliendo consistentemente con el umbral de $\text{IoU} \geq 0.75$ requerido
- Los patrones de pérdida muestran inestabilidad significativa, especialmente en la pérdida de clasificación y localización
- Se observa un probable sobreajuste del modelo, evidenciado por:
 - Alta variabilidad en las pérdidas de clasificación
 - Aumento significativo de la pérdida total al final del entrenamiento
 - Discrepancia entre métricas de evaluación y rendimiento real

Por lo tanto, a pesar de las métricas aparentemente positivas, el modelo no cumple con los criterios técnicos establecidos y no es confiable para su implementación en un entorno de producción. Se requieren ajustes significativos, incluyendo:

- Revisión de los datos de entrenamiento
- Ajuste de hiper parámetros
- Fortalecimiento de las técnicas de regularización
- Posible reconsideración de la arquitectura del modelo

Conclusión Modelos de Detección de Vacas

Como se mencionó en la guía de iteraciones, los modelos para detección de vacas se realizaron de forma paralela y esta sección compara los resultados finales de ambos modelos.

Modelo	Precisión	mAP	¿Los resultados son confiables?	¿Se cumplen los objetivos de negocio?
A1 Modelo de Detección con Pytorch	98.7%	0.975	Sí. En las imágenes de prueba que se le pasó al modelo entrenado detectó correctamente todas las vacas en las imágenes	Sí

D1 Modelo de Detección con Tensor Flow	NA	1.000	No. El modelo tuvo un pobre desempeño al identificar las vacas en las imágenes de prueba que se le pasaron.	No
--	----	-------	---	----