

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Querétaro

TC3007C

Inteligencia artificial avanzada para la ciencia de datos II

MODELO DE CLASIFICACIÓN DE POSICIONES PARA GANADO BOVINO (1ST VERSION- TENSORFLOW)

Autores:

A01368818 Joel Sánchez Olvera

A01661090 Juan Pablo Cabrera Quiroga

A01704076 Adrián Galván Díaz

A01708634 Carlos Eduardo Velasco Elenes

A01709522 Arturo Cristián Díaz López

Índice

Índice.....	1
Abstracto	1
Técnica de Modelado	2
Arquitectura.....	2
Supuestos de Modelado	2
Diseño de las Pruebas	3
Preprocesamiento de las Imágenes.....	3
Método de Entrenamiento.....	4
Ajuste de Parámetros	4
Descripción del Modelo	5
1. Modelo y Propósito.....	5
2. Precisión y Robustez Esperada.....	5
3. Interpretación del Modelo y Dificultades.....	5
4. Parámetros.....	5
5. Descripción Técnica del Modelo.....	6
6. Resultados y Patrones Detectados.....	7
7. Comportamiento del Modelo.....	7
8. Interpretación del Comportamiento del Modelo.....	8
Evaluación del Modelo	8
Resultados Principales.....	8
Parámetros revisados y ajustados	10

Abstracto

Este reporte detalla el desarrollo de un modelo de clasificación de imágenes basado en una red neuronal convolucional (CNN), implementada utilizando TensorFlow, la cual fue desarrollada para identificar y clasificar de los datos entrantes que son imágenes de vacas, desde un ángulo superior, en dos categorías: "**vaca acostada**" y "**vaca de pie**". A partir de un dataset de imágenes previamente clasificadas y organizadas en carpetas, aplicamos técnicas de preprocesamiento y ajuste de pesos por clase para manejar el desbalance en los datos y optimizar el rendimiento del modelo.

Técnica de Modelado

Arquitectura

Modelo: Red Neuronal Convolucional (CNN) en TensorFlow

Como mencionamos anteriormente el modelo del proyecto se basó en una red neuronal convolucional implementada con TensorFlow, la meta principal del modelo es clasificar correctamente una imagen dentro de las dos categorías mencionadas : **“vaca de pie”** y **“vaca acostada”**, para cumplir con dicha meta

Optimización y Entrenamiento

- Optimizador Adam
- Learning Rate 0.001.
- Función de pérdida: Binary Cross Entropy.

Supuestos de Modelado

Para éste modelo de clasificación, los supuestos principales se refieren a la calidad de las imágenes y las circunstancias de iluminación en las que se encuentran, ya que dado a que las fotos se toman cada 5 minutos a cualquier hora del día, tenemos diferentes condiciones de Iluminación en el dataset:

- **Variaciones de Iluminación:** Dentro del dataset, tenemos imágenes tomadas en la mañana del día que aparecen tener una saturación de color muy alta pero en realidad es debido a la luz de la imagen, así como fotos de noche que son muy oscuras y poco distinguibles.
- **Brillo de las imágenes:** A pesar del desbalance natural entre clases y las condiciones de iluminación, las imágenes seleccionadas para el modelo representan principalmente condiciones de iluminación diurna.
- **Desbalance en las Clases:** La categoría "vaca acostada" contiene más ejemplos que "vaca de pie". Para mitigar este desbalance implementamos un tope de imágenes por clase, asegurando que tenían una participación equitativa de 280 imágenes por clase.

Diseño de las Pruebas

Para evaluar de manera efectiva el desempeño del modelo, se implementó un enfoque sistemático basado en la división del dataset, el preprocesamiento de imágenes y el uso de métricas de evaluación.

División del Dataset

Las proporciones para la división del dataset, fueron las siguientes:

1. **Conjunto de Entrenamiento (80%):**

- Este conjunto se utilizó para ajustar los pesos del modelo mediante retropropagación, permitiendo al modelo aprender patrones clave de las imágenes.

2. **Conjunto de Prueba (20%):**

- Reservado para la evaluación final del modelo.
- Este conjunto mide la capacidad del modelo para generalizar en datos no vistos anteriormente.

Preprocesamiento de las Imágenes

Cada imagen pasó por un proceso de preprocesamiento antes de ser utilizada en el modelo. Esto incluyó:

1. **Redimensionamiento:**

- Las imágenes originales fueron escaladas a un tamaño estándar de **224x224 píxeles**.

2. **Normalización:**

- Los valores de los píxeles se ajustaron al rango [0,1] para estabilizar el entrenamiento y acelerar la convergencia.

3. **Aumentos de Datos (Data Augmentation):**

- Para incrementar la diversidad del dataset y mejorar la capacidad del modelo para generalizar, se aplicaron transformaciones como:
 - **Rotaciones aleatorias**
 - **Volteos horizontales**
 - **Cambios en el brillo y contraste**
 - **Zoom aleatorio**

Método de Entrenamiento

El modelo fue entrenado utilizando un esquema iterativo por **8 épocas**, con un tamaño de lote (**batch size**) de **32 imágenes** por iteración. La función de pérdida utilizada fue **Binary Cross Entropy**, optimizada mediante el algoritmo **Adam**.

Durante el entrenamiento, se monitorearon métricas clave en el conjunto de validación, incluyendo:

- Precisión (Accuracy).
- Pérdida (Loss).

Ajuste de Parámetros

Parámetros:

1. **img:224**

- a. Se eligió un tamaño de imagen de 224 x 224 píxeles para el procesamiento del modelo. Este tamaño es estándar en arquitecturas de CNNs, permiten un equilibrio entre la preservación de detalles esenciales en las imágenes y el uso de memoria y tiempo de procesamiento.

2. **batchsize:32**

- a. Se utilizó un tamaño de lote (batch size) de 32 imágenes por iteración. Este valor nos ayuda a tener un equilibrio entre la demanda del aprendizaje y el uso eficiente de la memoria.

3. **epochs: 8**

- a. El modelo entrenó durante 8 épocas, este número fue seleccionado para proporcionarle el tiempo suficiente de exposición a los datos, sin incurrir en overfitting. La decisión de usar este valor fue tomada en base al monitoreo de la precisión y pérdida del modelo conforme fue avanzando en el entrenamiento.

4. **device: GPU (CUDA, MPS)**

- a. El entrenamiento del modelo se realizó en la GPU utilizando CUDA Y MPS, lo que aceleró significativamente el entrenamiento sobre los datos y la optimización de los pesos.

5. **Learning Rate:0.001**

- a. El Learning Rate fue configurado en 0.001. Este valor se seleccionó para garantizar que el modelo se actualiza y ajusta de manera estable, observamos que fue el Learning Rate con mejor resultado.

Descripción del Modelo

1. Modelo y Propósito

El modelo desarrollado es una **Red Neuronal Convolutiva (CNN)** implementada en TensorFlow, diseñada específicamente para clasificar imágenes aéreas de vacas en dos categorías: "vaca acostada" y "vaca de pie". Este modelo tiene aplicaciones prácticas en la solución del reto pues nos da una clasificación precisa para usarse en el sistema de CAETEC.

2. Precisión y Robustez Esperada

Durante el entrenamiento, se utilizaron configuraciones específicas, como un tamaño de imagen de **224 x 224 píxeles**, un **batch size de 32** y un entrenamiento por **8 épocas**, para optimizar el modelo para este conjunto de datos. La arquitectura de la CNN fue seleccionada por su capacidad para manejar variaciones moderadas en las imágenes, como cambios en la iluminación y el contraste, con la arquitectura mencionada nos

aseguramos de que el modelo puede reconocer las imágenes en diferentes condiciones y la robustez del modelo y su resistencia a cambios es buena.

3. Interpretación del Modelo y Dificultades

La CNN fue diseñada para extraer características relevantes, como bordes, texturas y formas, que permitan clasificar con precisión la posición de las vacas. Sin embargo, debido a la complejidad del modelo, tuvimos que realizar ajustes a los hiperparámetros como el Learning Rate y los pesos que se le daban a cada clase para reconocerla de manera más precisa.

Características Útiles del Modelo

Algunas características destacadas de la implementación incluyen:

- **Data Augmentation:** Técnicas como rotación, cambio de brillo y volteo horizontal ayudaron a incrementar la diversidad de los datos.
- **Regularización con Dropout:** Evitó el sobreajuste al introducir aleatoriedad en la activación de neuronas durante el entrenamiento.

4. Parámetros

El modelo fue entrenado utilizando los siguientes hiperparámetros:

- **Tasa de aprendizaje:** Se configuró una tasa inicial de 0.001 utilizando el optimizador Adam.
- **Épocas de entrenamiento:** Se establecieron 8 épocas, suficientes para alcanzar la convergencia sin incurrir en overfitting.
- **Tamaño del lote (*batch size*):** Se utilizó un tamaño de lote de 32 imágenes por iteración.
- **Tamaño de imagen:** 224 x 224 píxeles.

5. Descripción Técnica del Modelo

La arquitectura del modelo fue definida como:

- **Capa de Entrada** (224x224x3)
- **1ra Capa Convolutiva** (32, 3x3, activación: ReLU)
- **1ra Capa de MaxPooling2d** (2x2)
- **2da Capa Convolutiva** (64, 3x3, activación: ReLU)
- **2da Capa de MaxPooling2d** (2x2)
- **3ra Capa Convolutiva** (128, 3x3, activación: ReLU)
- **3ra Capa de MaxPooling2d** (2x2)

- **Capa Flatten**
- **1ra Capa Densa** (128, activación: ReLU, dropout: 0.5).
- **Capa de Salida** (1, activación: Sigmoid).

Capa de Entrada:

- Dimensiones de las imágenes: 224x224x3.
- Preprocesamiento: Escalado de los valores de los píxeles al rango [0,1].

Capas Convolucionales:

- **Primera Capa:** 32 filtros con un tamaño de núcleo de 3x3, activación ReLU.
- **Segunda Capa:** 64 filtros con un tamaño de núcleo de 3x3, activación ReLU.
- **Tercera Capa:** 128 filtros con un tamaño de núcleo de 3x3, activación ReLU.

Capas de Agrupación (MaxPooling2d):

- Reducen la dimensionalidad mediante ventanas de 2x2, preservando las características más importantes y disminuyendo la carga computacional.

Capa Flatten:

- Convierte las matrices resultantes en vectores unidimensionales, preparando los datos para la etapa de clasificación.

Capas Densas:

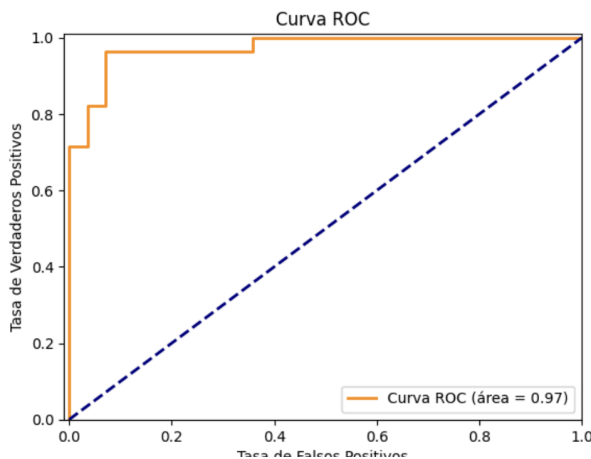
- Una capa densa completamente conectada con 128 unidades, activación ReLU, y regularización mediante Dropout (tasa de 0.5).
- Capa de salida con una unidad y activación Sigmoid, adecuada para tareas de clasificación binaria.

6. Resultados y Patrones Detectados

Los resultados obtenidos durante el entrenamiento y prueba destacan el desempeño robusto del modelo:

- **Precisión (P):** 96%.
- **Exhaustividad (Recall):** 96%.
- **F1-Score promedio:** 96%.

Curva ROC



La curva ROC muestra un área bajo la curva (AUC) de 0.97, indicando que el modelo tiene una alta capacidad para distinguir entre las dos clases, minimizando falsos positivos y falsos negativos.

7. Comportamiento del Modelo

1. **Precisión:** El modelo demuestra una alta capacidad para clasificar correctamente ambas categorías, con un 96% de precisión promedio.
2. **Recall:** Se asegura de identificar la mayoría de los casos relevantes, evitando omisiones críticas.
3. **Robustez :** Las métricas indican que el modelo no depende exclusivamente de patrones específicos del conjunto de entrenamiento, sino que generaliza bien a nuevas imágenes.

8. Interpretación del Comportamiento del Modelo

8.1 Precisión (Precision)

El modelo alcanzó una precisión promedio de **96%**, lo que indica su efectividad para minimizar los falsos positivos. Esto es crucial en el contexto del proyecto. El modelo es confiable para identificar correctamente a las vacas en sus respectivas categorías.

8.2 Exhaustividad (Recall)

Con un recall promedio de **96%**, el modelo demostró una alta capacidad para detectar todas las instancias relevantes en ambas categorías. El modelo tiene un bajo nivel de omisión de vacas en cualquiera de las categorías, lo que lo hace adecuado para aplicaciones de monitoreo continuo.

8.3 F1-Score

El F1-Score promedio de **96%** refleja un balance óptimo entre precisión y recall. Este equilibrio demuestra que el modelo no está sesgado hacia una categoría específica.

8.4 Curva ROC

La **Curva ROC** alcanzó un área bajo la curva (**AUC**) de **0.97**, lo que resalta la capacidad del modelo para distinguir correctamente entre las dos categorías.

Evaluación del Modelo

Resultados Principales

Precisión Global

- **Valor:** 96.43%.
- **Interpretación:**
 - El modelo logró clasificar correctamente el 96.43% de las imágenes del conjunto de prueba, confirmando un alto nivel de confiabilidad para esta tarea.

Pérdida

- **Valor Final en Prueba:** 0.1661.
- **Interpretación:**
 - Una pérdida baja indica que las predicciones del modelo se aproximan con precisión a las etiquetas reales, lo que sugiere que los parámetros ajustados y las configuraciones del modelo son apropiados.

Reporte de Clasificación

Clase	Precisión	Recall	F1-Score	Soporte
Acostada	96%	96%	96%	28
Parada	96%	96%	96%	28
Macro avg	96%	96%	96%	56

Weighted avg	96%	96%	96%	56
---------------------	------------	------------	------------	-----------

Ambas clases tienen un desempeño equivalente, lo que sugiere que el modelo no tiene sesgo hacia ninguna categoría.

El modelo desarrollado alcanzó un buen desempeño en la clasificación de imágenes de vacas, cumpliendo con los objetivos planteados. Las métricas obtenidas reflejan su robustez y capacidad para generalizar.

Parámetros revisados y ajustados

Durante el proceso de entrenamiento y ajuste del modelo CNN, se realizaron diversas modificaciones a los hiperparámetros con el objetivo de mejorar su desempeño y garantizar su capacidad de generalización. La **tasa de aprendizaje** inicial de 0.01 fue reducida a 0.001 para estabilizar las actualizaciones de los pesos y mejorar la convergencia. Asimismo, se incrementó la **tasa de dropout** de 0.3 a 0.5 en las capas densas finales, lo que permitió reducir el riesgo de overfitting.

Otro aspecto clave fue el ajuste de los **pesos por clase** para tratar el desbalance entre "vaca acostada" y "vaca de pie", y limitamos las imágenes a un cierto número de imágenes por clase, lo que garantizó que ambas categorías tuvieran una representación equitativa en el cálculo de la pérdida. Por otro lado, el número de **épocas** fue reducido de 10 a 8, ya que se observó que las métricas principales se estabilizaron alrededor de la sexta época, evitando así el sobreajuste.

Para mejorar la robustez del modelo frente a variaciones en las imágenes, se implementaron técnicas avanzadas de **data augmentation**, como rotaciones aleatorias en un rango de $[-15^\circ, 15^\circ]$, cambios de brillo y contraste, y volteos horizontales. Estas transformaciones incrementaron la diversidad del dataset, simulando condiciones reales de captura. Adicionalmente, el modelo fue ajustado para procesar imágenes de **224 x 224 píxeles**, en lugar de 128x128, lo que permitió preservar detalles visuales importantes para la clasificación.