Tecnológico de Monterrey

Campus Querétaro

Modelado #2

Gamaliel Marines Olvera	A01708746
Uri Jared Gopar Morales	A01709413
José Antonio Miranda Baños	A01611795
María Fernanda Moreno Gómez	A01708653
Oskar Adolfo Villa López	A01275287
Luis Ángel Cruz García	A01736345

Inteligencia artificial avanzada para la ciencia de datos II
Grupo 501

Introducción

En este documento se presenta el análisis y desarrollo de un modelo de clasificación de imágenes basado en redes neuronales convolucionales (CNN), aplicado a la identificación de tres clases: "cama vacía," "vaca acostada," y "vaca de pie." Este trabajo forma parte de un proyecto destinado a mejorar la automatización y precisión en la monitorización de camas para ganado, utilizando inteligencia artificial como herramienta principal.

El propósito principal de este documento es detallar las mejoras implementadas al modelo original (Modelo 1) y los resultados obtenidos tras la optimización. Durante la primera fase de desarrollo, el Modelo 1 mostró resultados prometedores, pero también expuso desafíos importantes, como la dificultad para clasificar correctamente clases minoritarias y la necesidad de un conjunto de datos más robusto para una mejor generalización.

Por ello, se desarrolló un segundo modelo (Modelo 2) que aborda estas limitaciones mediante ajustes clave en las transformaciones de imágenes, estrategias de entrenamiento y métodos de evaluación. Estas mejoras incluyen la incorporación de nuevas técnicas de preprocesamiento, como el zoom out para simular distintas alturas de cámara, el ajuste de pesos en la función de pérdida para combatir el desbalance de datos y la optimización del modelo en GPU para acelerar el entrenamiento. En adición a las mejoras, se implementaron nuevas métricas de evaluación del modelo como lo es la gráfica de accuracy (precisión) por época y el loss (pérdida) por época igualmente

Técnica de modelado

La técnica de modelado del Modelo 2 no cambió con respecto a la del Modelo 1, a continuación, se presenta la técnica de modelado descrita en el Modelo 1.

Utilizamos una técnica de Red Neuronal Convolucional (CNN) debido a su eficacia para procesar datos en forma de cuadrícula, como las imágenes. La CNN está formada por varias capas: capas convolucionales, capas de agrupación y capas totalmente conectadas. Esta arquitectura se asemeja al procesamiento visual del cerebro humano y es adecuada para capturar patrones jerárquicos y dependencias espaciales dentro de imágenes.

Las capas que se utilizaron para esta arquitectura fueron las siguientes:

- **Capas convolucionales:** Aplican operaciones convolucionales a las imágenes. Sirven para detectar bordes, texturas y patrones complejos.

- **Capas de agrupación**: Reducen el tamaño de las dimensiones espaciales de la entrada con el fin de reducir la complejidad computacional.
- **Funciones de activación:** Utilizamos Rectified Linear Unit (ReLU) para introducir una propiedad no lineal al modelo con el fin de aprender patrones más complejos, evita el vanishing gradient[1] al no saturarse de entradas positivas, mitigando este problema y acelerando la convergencia durante el entrenamiento, además que es computacionalmente más eficiente que otras funciones al tener una forma matemática directa.
- **Capas completamente conectadas:** Estas son las responsables de hacer las predicciones basadas en los aprendizaes de las capas anteriores.

Las capas son entrenadas por medio de carpetas con imágenes clasificadas de acuerdo a las etiquetas de “cama vacía”, “vaca acostada” y “vaca de pie”, por lo que la red aprende a reconocer patrones y características asociadas a cada clase.

Para mejorar la generalización del modelo, se implementó la técnica de aumento de datos (data augmentation), que es el proceso de generar artificialmente nuevos datos a partir de datos existentes para entrenar un modelo, ya que los modelos de Machine Learning requieren de conjuntos de datos reales diversos y bastos para la generalización del modelo. En nuestro modelo, se realizaron las siguientes aumentaciones: redimensionamiento, giro horizontal y vertical de la imagen y rotación, esto con el fin de aumentar artificialmente los datos y tener un conjunto de datos de entrenamiento más robusto para que pueda realizar mejores predicciones el modelo.

Suposiciones

Las suposiciones, de igual manera, son las mismas del Modelo 1. A continuación, se muestran las suposiciones del modelo descritas en el Modelo 1:

- Las imágenes de las camas tienen el mismo tamaño, 950 x 450 px.
- Las imágenes muestran una sola cama.
- Las imágenes no muestran personas u objetos diferentes.
- Las imágenes de las camas se toman desde arriba.
- Las imágenes fueron tomadas por un modelo específico de cámara y fueron pasadas por un proceso de aplanado, que coincide con el ojo de pescado[2] generado por dicho tipo de cámara en específico.

[1] **Vanising gradient:** Es un problema que ocurre en los entrenamientos de las CNN cuando los gradientes de la función de pérdida con respecto a los pesos de las primeras capas se vuelven extremadamente pequeños, resultando que las capas reciban poca o nula información de los pesos en la retropropagación (backpropagation).
[2] **Ojo de pescado:** Se refiere a las distancias focales que tienen ángulos muy anchos y que consiguen producir una distorsión visual de grandes capacidades.

Diseño de pruebas

Se utilizó el diseño de pruebas descrito en el Modelo 1. A continuación, se inserta el diseño de pruebas descrito en el Modelo 1:

Para las pruebas se utilizará el conjunto de datos que fue seleccionado para dicho propósito desde un inicio, lo que brinda la posibilidad de ver el rendimiento del modelo en imágenes que no fueron utilizadas para su entrenamiento, logrando de esta forma identificar si existe un sobreajuste[3].

Específicamente, se cuenta con la siguiente cantidad de imágenes por cada clase para el conjunto de prueba:

- Cama vacía: 1323
- Vaca acostada: 394
- Vaca de pie: 60

Una vez utilizado el modelo para evaluar los nuevos datos, se obtendrá como métrica principal el accuracy en forma de porcentaje, el cual se obtiene implementando la siguiente fórmula:

$$Accuracy = \frac{Correct\ predictions}{Total\ predictions} * 100$$

Buscando obtener un valor que no solo muestre que el ajuste del modelo es correcto, sino también que no existe una diferencia significativa entre dicho accuracy y el obtenido durante el entrenamiento.

Por último, se utilizará una matriz de confusión para poder identificar si existe tendencia a predecir una clase en específico, el cual es otro problema que se busca evitar, ya que al contar con pocas imágenes de “Vaca de pie”, podría no estar detectando correctamente dicha clase y seguir mostrando un accuracy alto.

Los valores que se muestran en la matriz de confusión son acorde a las clases, en donde se comparará la cantidad de valores predecidos correctamente para cada clase. A continuación se muestra un diagrama que ejemplifica los valores obtenidos en una matriz de confusión, en dicho diagrama, los valores que se deberían mostrar más altos son los verdaderos positivos (TP) y los verdaderos negativos (TN), ya que significa que el número de predicciones por cada clase fueron correctas.

[3] **Sobreajuste (overfitting):** Es un comportamiento en Machine Learning (ML) que se produce cuando el modelo da predicciones precisas para los datos de entrenamiento, pero no para nuevos datos, es decir, aprende de memoria los resultados, sin embargo, no aprende los patrones en realidad para datos no vistos.

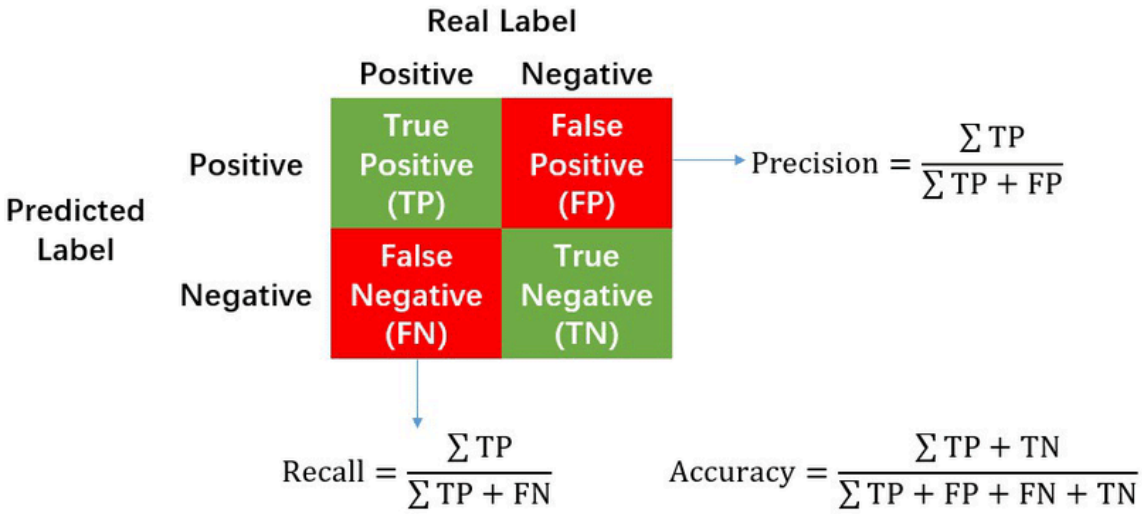

Imagen 1: Diagrama de construcción de una matriz de confusión.

Modelo

Diseño y configuración del segundo modelo

En el segundo modelo, se utilizaron las mismas métricas implementadas en el Modelo 1 para evaluar el conjunto de prueba, con la excepción de la curva ROC, que fue omitida en esta iteración. Las métricas y visualizaciones utilizadas fueron:

1. **Accuracy:** Continuó siendo una métrica importante para medir el rendimiento general, aunque se complementó con análisis más específicos para entender mejor el desempeño en clases minoritarias.
2. **Gráficos de pérdida (loss) y precisión (accuracy) por época:** Permiten observar la evolución del modelo durante las 50 épocas de entrenamiento, asegurando que el modelo no se sobreajuste y que generalice mejor.
3. **Matriz de confusión:** Se mantuvo como una herramienta clave para analizar la distribución de las predicciones y detectar tendencias de confusión entre clases, especialmente en las minoritarias.

La curva ROC, aunque útil en problemas binarios o datasets balanceados, mostró limitaciones en su capacidad para reflejar de manera precisa el desempeño en un problema multiclase con datos desbalanceados como este. Dado que las clases "vaca acostada" y "vaca de pie" tienen menos representaciones, el AUC puede dar una falsa sensación de buen rendimiento global, ocultando las deficiencias del modelo en las clases menos representadas. Por ello, se decidió reemplazar el uso de la curva ROC por un enfoque más centrado en la matriz de confusión y las

métricas específicas por clase, como precision y recall, que permiten un análisis más detallado del comportamiento del modelo frente a datos desbalanceados.

Transformaciones de imágenes

A las transformaciones originales del Modelo 1 (redimensionamiento, giros horizontales y verticales aleatorios, y rotación aleatoria), se añadió la transformación de **zoom out** mediante *RandomAffine* con escalado en el rango de (0.5, 1.0). Esta transformación emula diferentes alturas desde las cuales podría capturarse la imagen, incrementando la diversidad del conjunto de datos de entrenamiento y mejorando la robustez del modelo frente a variaciones espaciales.

Entrenamiento

- **Pesos por Clase**: Debido al desbalance en la cantidad de imágenes por clase, se introdujeron pesos específicos en la función de pérdida *CrossEntropyLoss*. Las clases minoritarias ("vaca acostada" y "vaca de pie") reciben un peso mayor (3.0 y 6.0, respectivamente), para reducir el sesgo hacia la clase más representada ("cama vacía").
- **Aumento de Épocas**: Se incrementó el número de épocas de entrenamiento de 5 a 50, permitiendo al modelo aprender patrones más complejos y mejorar su capacidad de generalización.
- **Optimización con GPU**: El modelo fue optimizado para entrenarse utilizando GPU (si es que está disponible), acelerando significativamente el proceso de entrenamiento y facilitando el manejo de un mayor número de épocas.
- **Historial de Entrenamiento**: Se implementó un registro visual del historial de pérdida durante el entrenamiento y del accuracy en el conjunto de validación, facilitando la evaluación continua del modelo y la detección temprana de sobreajuste.
- **Almacenamiento Dinámico de Modelos**: El modelo se guarda automáticamente cuando supera un umbral de accuracy en validación (95% inicialmente), actualizándose con cada mejora en el rendimiento.

Métricas y pruebas en el conjunto de Test (Prueba)

Se introdujo la generación de una matriz de confusión como herramienta adicional para evaluar el rendimiento del modelo en el conjunto de prueba. Este análisis proporciona información más detallada sobre posibles tendencias hacia la predicción incorrecta de clases minoritarias, especialmente la clase "vaca de pie."

Evaluación del segundo modelo

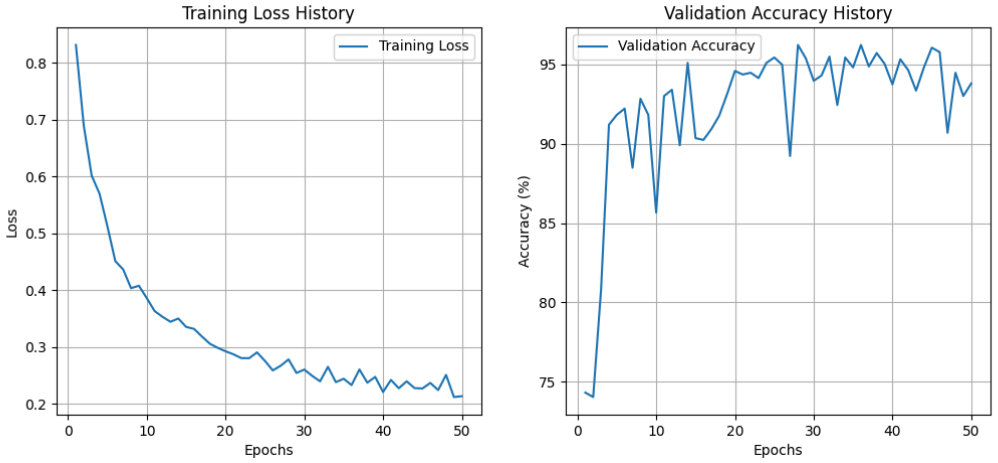

Imagen 2: Gráfico de pérdida de training y de precisión en el conjunto de validación.

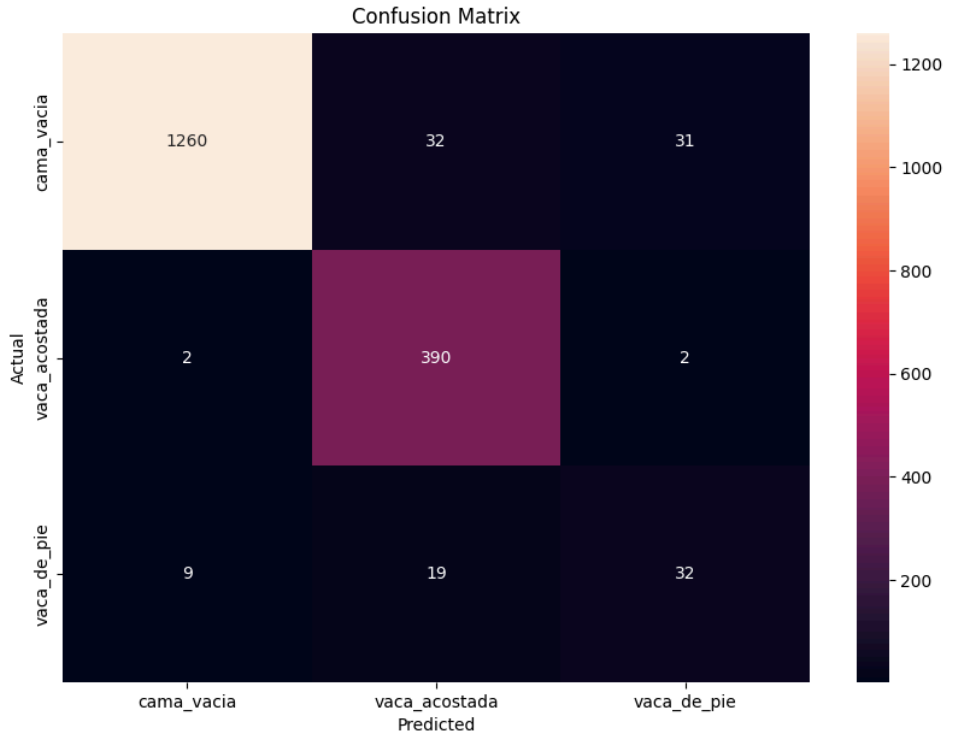

Imagen 3: Matriz de confusión del segundo modelo.

El Modelo 2 fue diseñado para superar las limitaciones identificadas en el Modelo 1, específicamente la dificultad para clasificar correctamente las clases minoritarias y la falta de robustez en las predicciones debido a la variabilidad de los datos de

entrenamiento. Tras implementar mejoras clave en el preprocesamiento, entrenamiento y evaluación, los resultados obtenidos muestran un avance significativo, aunque aún hay áreas de mejora.

Los gráficos de pérdida y precisión (**Imagen 2**) por época para el Modelo 2 reflejan una mejora consistente en comparación con el Modelo 1.

- **Gráfico de pérdida:** Durante las 50 épocas de entrenamiento, se observa una disminución sostenida en la pérdida, indicando que el modelo fue capaz de optimizar sus pesos y minimizar errores. Esto es un indicador de que el aumento en las épocas permitió al modelo aprender patrones más complejos en los datos.
- **Gráfico de precisión:** La precisión (accuracy) alcanzó un valor superior al 96% al final del entrenamiento, con variaciones mucho más estables en comparación con el Modelo 1. Esto sugiere que las nuevas transformaciones y el ajuste de pesos por clase ayudaron al modelo a generalizar mejor. El Modelo 2 obtuvo una precisión de 96.22%, superando los resultados del Modelo 1. Esto valida la efectividad de las transformaciones adicionales y del ajuste en los pesos en la función de pérdida.
- **Matriz de confusión (Imagen 3):**
    - *Cama vacía:* Esta clase se mantuvo con un alto nivel de precisión, con la mayoría de las imágenes clasificadas de manera correcta.
    - *Vaca acostada:* Esta clase incrementó su número de predicciones correctas, reduciendo así la confusión con otras clases.
    - *Vaca de pie:* En esta clase sí hubo mejora en el número de imágenes clasificadas de manera correcta, sin embargo, aún persisten errores al confundir imágenes con las clases “cama vacía” y “vaca acostada”, sugiriendo que aun el desbalance de los datos sigue siendo un factor importante, a pesar del ajuste que se hizo en los pesos.

Tabla comparativa entre el Modelo 1 y el Modelo 2

Aspecto	Modelo 1	Modelo 2
Transformaciones en imágenes	Redimensionamiento, giros horizontales y verticales, rotación aleatoria.	Transformaciones del Modelo 1 + Zoom out
Épocas de entrenamiento	5 épocas	50 épocas
Pesos en función de pérdida	No se utilizaron pesos específicos para las clases.	Pesos ajustados: 3.0 para "vaca acostada" y 6.0 para "vaca de pie"

		debido al desbalance de clases.
Learning rate	0.001	0.001
Desempeño en la clase “cama vacía”	Excelente desempeño, con la mayoría de las imágenes clasificadas correctamente.	Excelente desempeño, sin cambios significativos respecto al Modelo 1.
Desempeño en la clase “vaca acostada”	Mejorable, con algunas confusiones hacia otras clases.	Mejora significativa, con menor cantidad de confusiones hacia otras clases.
Desempeño en la clase “vaca de pie”	Desempeño limitado, con alta frecuencia de confusiones con "cama vacía" y "vaca acostada".	Mejora moderada, aunque persisten errores debido al desbalance de datos.
Matriz de confusión	Mayor precisión en clases mayoritarias ("cama vacía"), pero menor en clases minoritarias.	Mejor distribución, aunque "vaca de pie" sigue siendo la clase más difícil de clasificar.
Áreas de mejora	Manejo del desbalance de datos, aumento de transformaciones y mayor número de épocas de entrenamiento.	Mejorar la detección de la clase "vaca de pie".
Accuracy en Test	95%	96.22%

En conclusión, el Modelo 2 representa un avance significativo en términos de precisión y generalización, superando los desafíos identificados en el Modelo 1. Sin embargo, el desempeño en clases minoritarias como "vaca de pie" evidencia la necesidad de seguir optimizando la estrategia de manejo de desbalance y de explorar otras soluciones para mejorar la capacidad del modelo en la identificación de patrones menos representados.

Análisis de datos

Al tener métricas de evaluación con resultados mejores al respecto del modelo 1 (*véase la Tabla Comparativa entre el Modelo 1 y el Modelo 2)*, procedimos en la búsqueda de hallazgos a partir de la utilización del modelo en las imágenes de entrenamiento y pruebas. Para esto, se utilizó un avance del código de despliegue, el cual recorta las imágenes en un número de camas definido previamente con una UI. Posteriormente, cada imagen individual es analizada por el modelo, y el

resultado de cada cama se almacena un archivo CSV con una etiqueta de la hora correspondiente, la cual se toma del nombre de la imagen.

Una vez teniendo el CSV con el uso de las camas, se pueden hacer gráficas para observar el uso de las camas.

Se realizó este procedimiento, pero se encontraron resultados que no coincidían con lo observado a simple vista. Por ejemplo, la cama de la derecha está vacía en una gran mayoría de las imágenes, pero en el análisis se obtuvo que era la cama más utilizada.

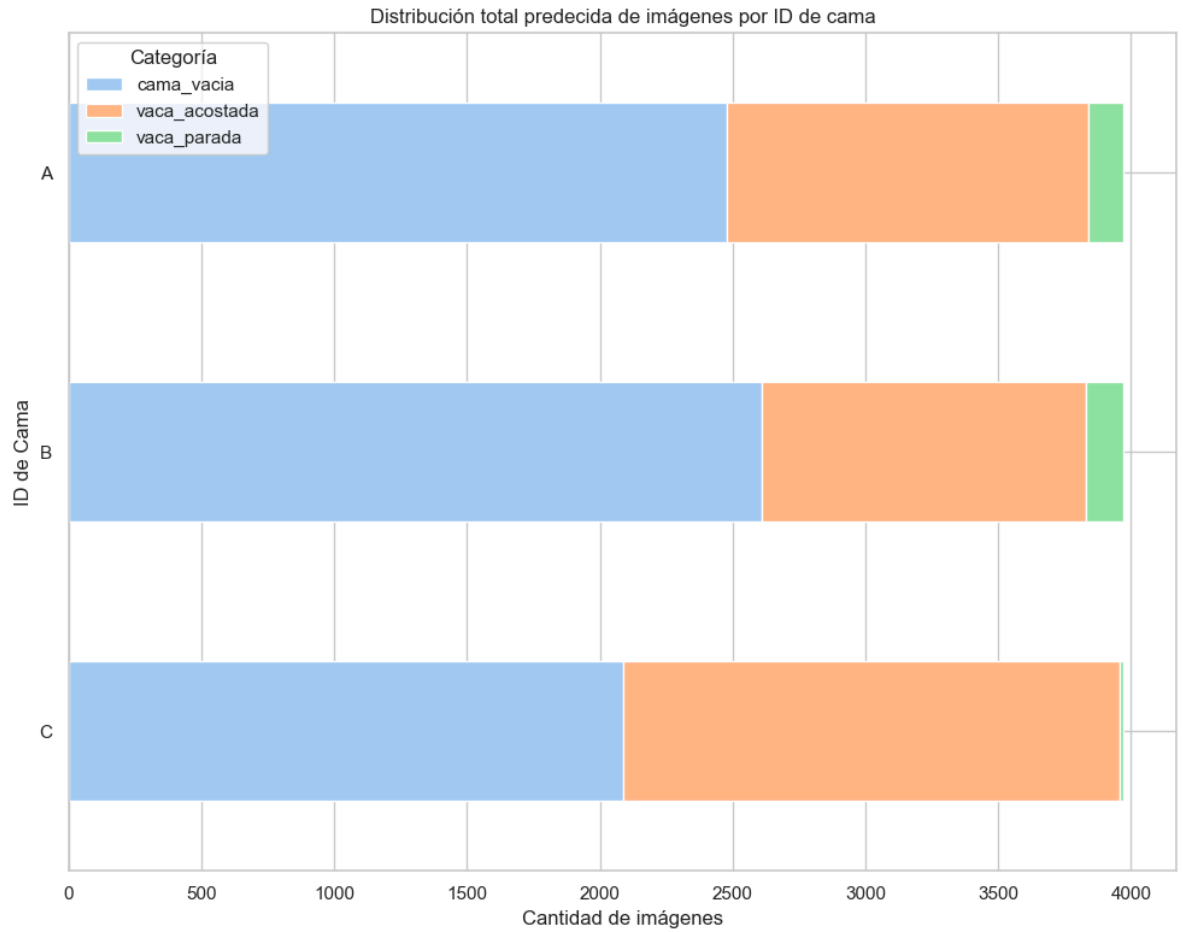

Imagen 4: Gráfico que muestra la cantidad de predicciones de las imágenes por cama.

Debido a esta inconsistencia, se decidió realizar una prueba adicional. Se graficó el uso de cada cama basado en el conjunto de entrenamiento clasificado manualmente, y se graficó también el uso predecido por el modelo sobre este mismo conjunto. Como se puede observar, los resultados difieren en gran magnitud,

lo que indica un error en el modelo. Es por ello que se decidió realizar una tercera iteración de modelado para explorar este problema.

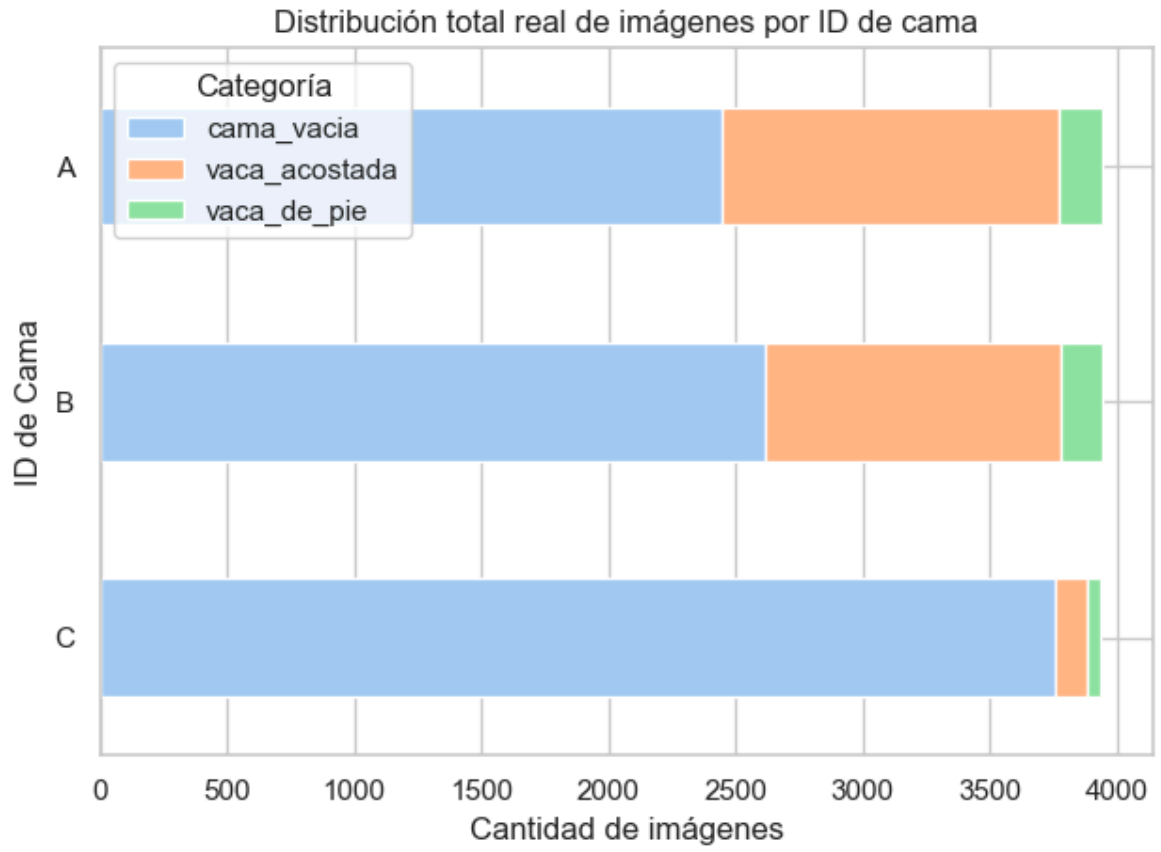

Imagen 5: Gráfico que muestra la distribución real de la cantidad de las imágenes por cama. Como podemos observar, en la cama C los resultados de las predicciones de dicha cama muestran más camas con vacas acostadas de las camas etiquetadas en la clase C.

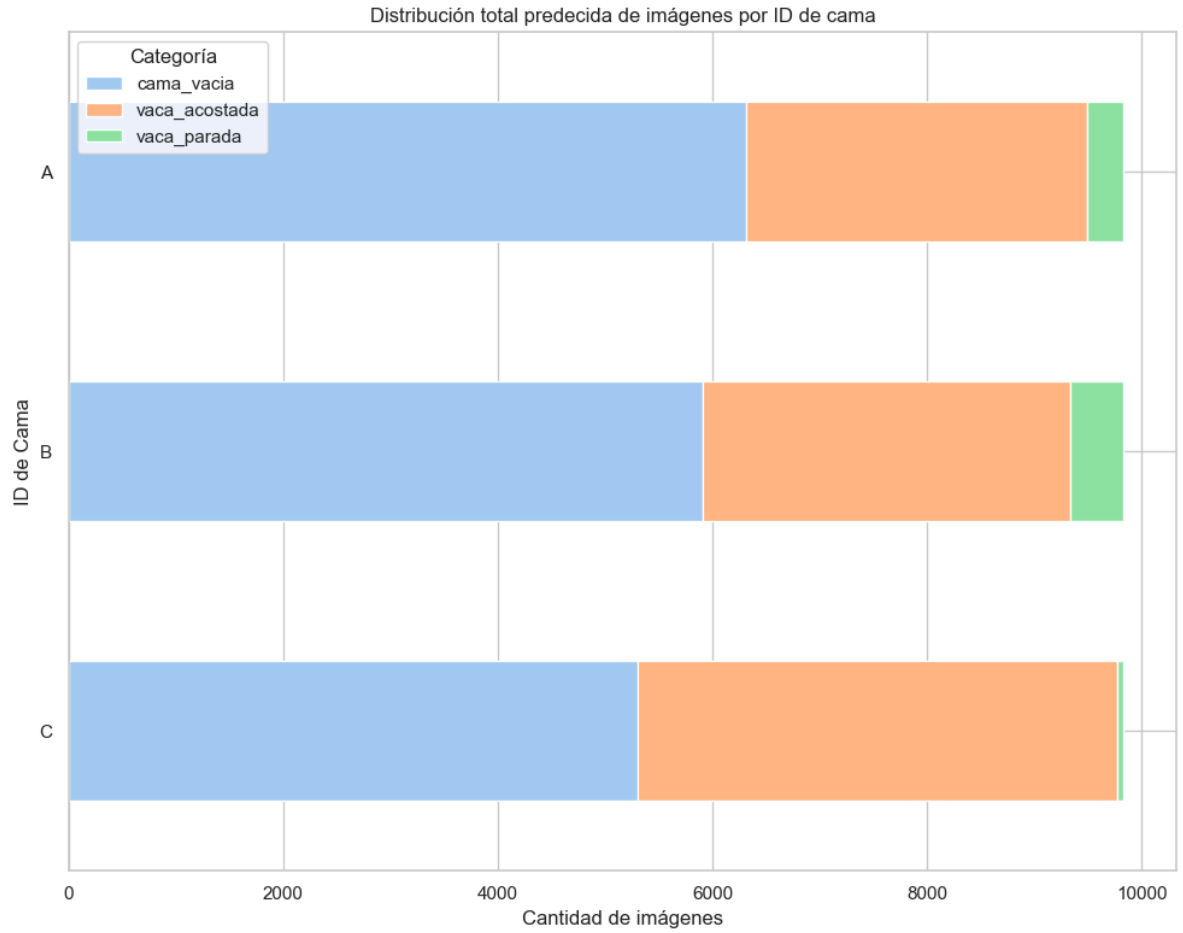

Imagen 6: Distribución de las imágenes de cada cama de acuerdo a las categorías. Dataset completo. Predicción del modelo.

La distribución de las predicciones presenta un sesgo significativo y resulta poco confiable al compararla con la muestra y con la distribución real. Por esta razón, se recomienda la construcción de un tercer modelo, ya que el modelo actual, a pesar de mostrar buenos resultados en las métricas de evaluación, no es adecuado para su implementación en producción. Esto se debe a que, particularmente en la cama "C," el modelo no clasifica correctamente los estados de las camas, generando un número mayor de imágenes etiquetadas como "vaca acostada" en comparación con la realidad, cuando estas imágenes deberían clasificarse como "cama vacía."

Este problema refleja la presencia de **falsos positivos**, es decir, resultados aparentemente buenos según las métricas tradicionales, que en la práctica no corresponden con el desempeño real del modelo. Estos falsos positivos generan

una falsa impresión de eficacia, evidenciando que el modelo no está cumpliendo adecuadamente con la tarea de clasificación.

Para abordar este desafío, se recomienda la incorporación de métricas adicionales que permitan evaluar de manera más precisa el desempeño del modelo en la clasificación de cada cama. Estas métricas deben enfocarse en reducir el desbalance en el accuracy por cada clase y garantizar un análisis más detallado del comportamiento del modelo. De esta manera, será posible seleccionar un modelo más robusto y confiable para su implementación en producción, que clasifique correctamente los tres estados de la cama con mayor precisión y equidad entre las clases.

Referencias al documento del Modelo 1

- [04]-Modeling_1_TC.pdf

Referencias

GeeksforGeeks. (n.d.). *Convolutional neural network (CNN) in machine learning*. Recuperado de https://www.geeksforgeeks.org/convolutional-neural-network-cnn-in-machine-learning/

Srivastava, S. (2023). *Understanding the difference between ReLU and Sigmoid activation functions in deep learning*. Medium. Recuperado de https://medium.com/@srivastavashivansh8922/understanding-the-difference-between-relu-and-sigmoid-activation-functions-in-deep-learning-33b280fc2071

Amazon Web Services (AWS). (n.d.). *What is data augmentation?* Recuperado de https://aws.amazon.com/what-is/data-augmentation/#:~=Data%20augmentation%20is%20the%20process,machine%20learning%20(ML)%20models

Amanatulla, M. (2023). *Vanishing Gradient Problem in Deep Learning: Understanding, Intuition, and Solutions.* Medium. Recuperado de https://medium.com/@amanatulla1606/vanishing-gradient-problem-in-deep-learning-understanding-intuition-and-solutions-da90ef4ecb54

Bargainfotos. (n.d.). *Todo sobre los objetivos de ojo de pez.* Recuperado de https://bargainfotos.com/blog/todo-sobre-los-objetivos-de-ojo-de-pez/

Amazon Web Services (AWS). (n.d.). *What Is Overfitting?* Recuperado de https://aws.amazon.com/es/what-is/overfitting/

Investopedia. (n.d.). *What Is Nonlinearity?* Recuperado de https://www.investopedia.com/terms/n/nonlinearity.asp

Sue, N. (2023). *What Is the Softmax Function Used in Deep Learning? Illustrated in an Easy-to-Understand Way.* Medium. Recuperado de https://medium.com/@sue_nlp/what-is-the-softmax-function-used-in-deep-learning-illustrated-in-an-easy-to-understand-way-8b937fe13d49

Analytics Vidhya. (2023). *What Is Adam Optimizer?* Recuperado de https://www.analyticsvidhya.com/blog/2023/09/what-is-adam-optimizer/