

Fecha de publicación 26 de mayo de 2024

## **Modelos de Machine Learning aplicados a la clasificación de imágenes de razas de ganado bovino**

V. Ortega Botero <sup>1</sup>

<sup>1</sup>Universidad de Antioquia, departamento de Ingeniería de Sistemas

### **ABSTRACT**

En este documento se presenta el proceso de desarrollo y evaluación de algunos modelos de aprendizaje automático para la clasificación de imágenes de razas de ganado bovino. La clasificación precisa de razas de ganado bovino a partir de imágenes es una tarea fundamental en diversos campos, como la ganadería y la gestión de inventarios. El enfoque propuesto utiliza técnicas de aprendizaje automático para extraer características relevantes de las imágenes, incluyendo redes neuronales convolucionales (CNN). El proceso incluye la recolección de un conjunto de datos con imágenes de diferentes razas de ganado bovino, la limpieza y preparación de dichos datos, la elección de los algoritmos de aprendizaje automático más adecuados y el entrenamiento de los modelos. Se exploran diferentes enfoques y se realiza la evaluación de los modelos utilizando métricas de desempeño. Los resultados muestran que los modelos de aprendizaje automático desarrollados logran un rendimiento satisfactorio en la clasificación de las razas de ganado bovino, demostrando su potencial para aplicaciones prácticas en el campo de la ganadería. Este estudio ofrece una base para futuras investigaciones en el desarrollo de modelos de aprendizaje automático para la clasificación de imágenes de razas de ganado bovino.

**PALABRAS CLAVE:** Aprendizaje automático, clasificación de imágenes, reconocimiento de razas de ganado, redes neuronales artificiales, machine learning, entrenamiento, modelos, validación, CNN, ganadería.

### **I. INTRODUCCIÓN**

En los últimos años, los avances en el campo del aprendizaje automático, especialmente en técnicas de deep learning, han permitido alcanzar resultados sobresalientes en tareas de clasificación de imágenes. Estos modelos son capaces de extraer características complejas y discriminativas de las imágenes, lo que ha impulsado el desarrollo de aplicaciones de clasificación en diversos dominios.

Sin embargo, la clasificación precisa de razas de ganado bovino en imágenes aún presenta desafíos debido a la variabilidad en la apariencia de las diferentes razas, las distintas perspectivas en las imágenes, los cambios de iluminación y la presencia de otros objetos en el entorno. Por lo tanto, se requiere un enfoque adecuado para abordar este problema y mejorar la precisión de la clasificación.

En este contexto, este artículo propone explorar y comparar varios modelos de machine learning

aplicados a la clasificación de imágenes de razas de ganado bovino. Se evaluarán diferentes enfoques, como redes neuronales convolucionales (CNN) y modelos pre-entrenados, con el objetivo de determinar cuál es el más eficiente y preciso para este problema específico.

## **II. EXPERIMENTOS**

En esta sección se describe la metodología de validación utilizada y la base de datos empleada en el proyecto. También se proporcionan detalles sobre las clases, el número de muestras, la distribución de muestras y las técnicas de balanceo inteligente.

### **A. BASE DE DATOS**

La base de datos utilizada en este proyecto consiste en un conjunto de imágenes de diferentes razas de ganado bovino. Las clases incluidas en el conjunto de datos son: Angus, Beefmaster, Bison, Brahman, Charolais, Guzerat, Holstein, Simmental

### **B. NÚMERO DE MUESTRAS Y DISTRIBUCIÓN**

El conjunto de datos se compone de un número total de muestras distribuidas de la siguiente manera:

Angus: 1000 imágenes

Beefmaster: 1000 imágenes

Bison: 1000 imágenes

Brahman: 1000 imágenes

Charolais: 1000 imágenes

Guzerat: 1000 imágenes

Holstein: 1000 imágenes

Simmental: 1000 imágenes

### **C. METODOLOGÍA DE VALIDACIÓN**

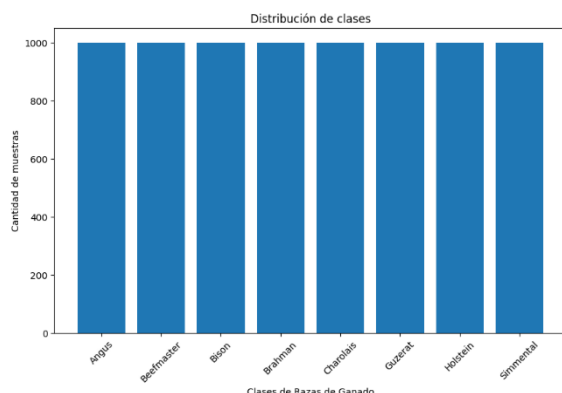
La metodología de validación utilizada es la validación cruzada estratificada. En este enfoque, se divide automáticamente el conjunto de datos en un conjunto de entrenamiento y un conjunto de validación. Al definir el generador de datos con ImageDataGenerator y establecer `validation_split=0.2`, se realiza una partición en la que el 80% de los datos se destina al conjunto de entrenamiento y el 20% al conjunto de validación. Esta estrategia garantiza que ambas divisiones mantengan una proporción equilibrada de muestras de cada clase, evitando así sesgos y asegurando una evaluación imparcial del modelo.

### **D. BALANCEO INTELIGENTE**

Para abordar el desequilibrio en la distribución de muestras entre las diferentes clases de razas de ganado, se emplearon técnicas de aumento de datos y submuestreo. Cada clase originalmente contenía 1000 imágenes. Para asegurar un conjunto de datos más equilibrado y robusto, se utilizó ImageDataGenerator para aplicar varias transformaciones a las imágenes, incluyendo rotaciones, traslaciones, cambios de brillo y volteo horizontal. Este proceso generó imágenes adicionales de las clases menos representadas, asegurando que cada clase mantuviera una cantidad adecuada de datos para el entrenamiento.

Por otro lado, aunque todas las clases originalmente tenían 1000 imágenes, se implementó un proceso para verificar y mantener esta cantidad, asegurando que no hubiera exceso ni déficit de muestras en ninguna clase. Este proceso fue visualizado a través de gráficos de barras que mostraban la distribución de las muestras por clase, confirmando que todas las clases contenían exactamente 1000 imágenes. En conjunto, estas técnicas de balanceo inteligente ayudaron a mejorar la equidad en el entrenamiento y evaluación del modelo, garantizando que todas las clases

estuvieran representadas de manera uniforme. Dando este resultado:



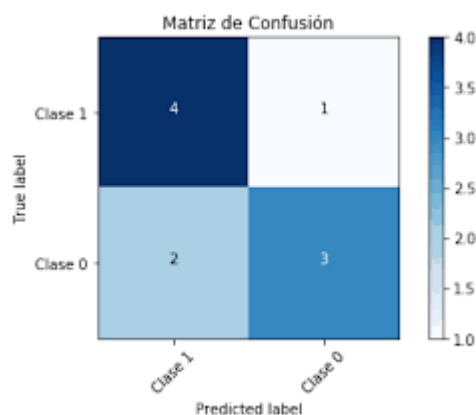
**Figura 1:** distribución del muestreo

La implementación inteligente de estas técnicas de submuestreo y sobremuestreo permite abordar de manera efectiva el desbalance en la distribución de muestras y obtener un conjunto de datos equilibrado para el proyecto. Esto brinda una base sólida para el entrenamiento y la evaluación de los modelos de reconocimiento de tipos de razas de ganado bovino, con resultados más precisos y confiables.

### III. MEDIDAS DE DESEMPEÑO

#### A. MATRIZ DE CONFUSIÓN

En problemas multiclase, la matriz de confusión es una representación tabular que muestra la relación entre las clases reales y las predicciones realizadas por el modelo para cada clase. A diferencia del caso binario, en un problema multiclase, la matriz de confusión tiene una estructura más compleja.



**Figura 2:** ejemplo matriz de confusión

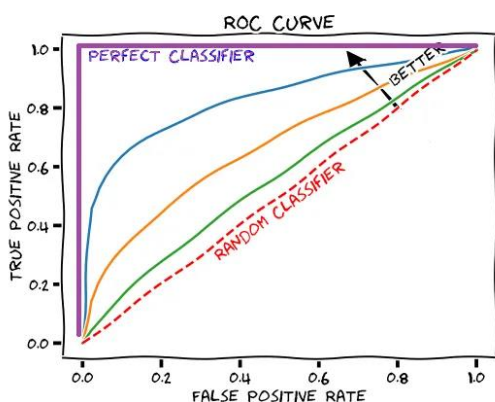
La matriz de confusión tiene dimensiones  $n \times n$ , donde  $n$  es el número de clases en el problema. Cada fila de la matriz representa la clase real, mientras que cada columna representa la clase predicha por el modelo. Cada celda de la matriz de confusión contiene el recuento de instancias que pertenecen a la clase real correspondiente (fila) y que fueron clasificadas en la clase predicha correspondiente (columna).

En un escenario ideal, todos los valores estarían en la diagonal principal de la matriz, lo que indicaría una clasificación perfecta para todas las clases. Sin embargo, en la práctica, pueden ocurrir errores de clasificación en forma de falsos positivos y falsos negativos para cada clase. Por lo tanto, la matriz de confusión multiclase permite analizar los aciertos y errores específicos para cada clase, proporcionando información detallada sobre el rendimiento del modelo en cada categoría. Esta información es fundamental para identificar patrones de errores y mejorar el modelo de clasificación.

#### B. CURVA ROC

La curva ROC (Receiver Operating Characteristic) es una herramienta gráfica empleada para evaluar el rendimiento de un modelo de clasificación, principalmente en problemas de clasificación binaria. No obstante, también se puede aplicar en problemas de clasificación multiclase generando curvas ROC para cada clase en comparación con el resto. Este enfoque proporciona una visión detallada del

rendimiento del modelo para cada clase individualmente, ayudando a comprender cómo se comporta el clasificador en diferentes escenarios de clasificación. Cada curva ROC representa la tasa de verdaderos positivos (sensibilidad) frente a la tasa de falsos positivos (1 - especificidad) para un umbral de decisión específico. Un área bajo la curva (AUC) cercana a 1 indica un mejor rendimiento del modelo, mientras que un valor cercano a 0.5 sugiere un rendimiento similar al azar. La interpretación de las curvas ROC en problemas multiclase puede ser más compleja que en problemas binarios, pero sigue siendo una herramienta valiosa para evaluar y comparar modelos de clasificación en diferentes configuraciones.

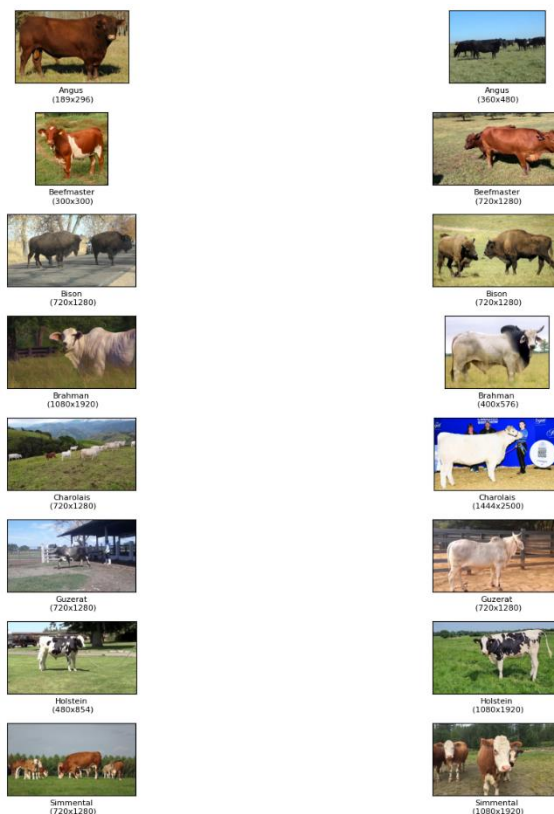


**Figura 3:** ejemplo de curva ROC

#### IV. PREPARACIÓN DEL DATASET

Antes de comenzar con la creación del modelo, es crucial preparar los datos de muestra para trabajar de manera más cómoda y efectiva. Para esto, es importante comprender la estructura y

características de los datos actuales.



**Figura 4:** imágenes originales

Para asegurar un procesamiento eficiente, estabilidad del modelo y una mejor capacidad de generalización, es fundamental normalizar las imágenes para que todas tengan el mismo tamaño y escala de colores RGB. Esto dará como resultado un dataset normalizado que facilitará el entrenamiento del modelo y mejorará su desempeño en datos nuevos.

La normalización de las imágenes implica redimensionarlas a un tamaño específico, lo que garantiza que todas tengan las mismas dimensiones. Además, se recomienda escalar los valores de píxeles al rango de  $[0, 1]$  dividiendo cada valor por 255, que es el valor máximo de intensidad para un canal de color RGB. Esto ayuda a estandarizar la escala de colores y facilita el proceso de aprendizaje para el modelo.

Al normalizar el dataset de imágenes, se crea una representación más consistente y homogénea de

[illegible]

Entonces durante el proceso de reescalado también se realiza la división del dataset en un conjunto de entrenamiento y un conjunto de validación, donde el 80% de las muestras se asignan al conjunto de entrenamiento y el 20% al conjunto de validación. Esta división asegura que el modelo se entrene con una cantidad suficiente de datos mientras se reserva una porción para evaluar su desempeño de manera independiente.

Con esta preparación completa, el dataset está listo para ser utilizado en la construcción y entrenamiento del modelo. Estas preparaciones son esenciales para asegurar que las imágenes se ajusten correctamente al modelo y puedan ser procesadas de manera eficiente durante el entrenamiento y la validación.

Es esencial definir métricas personalizadas para evaluar el rendimiento de un modelo de aprendizaje automático, especialmente en problemas de clasificación. Estas métricas, como

La sensibilidad, o recall, indica la proporción de casos positivos que fueron correctamente identificados por el modelo. La precisión, por otro lado, representa la proporción de casos identificados correctamente como positivos entre todos los casos identificados como positivos por el modelo. La puntuación F1 es una medida combinada de sensibilidad y precisión que proporciona una evaluación más equilibrada del rendimiento del modelo.

## VI. MODELO: REDES NEURONALES ARTIFICIALES

La CNN está diseñada para procesar imágenes de entrada de tamaño 64x64 píxeles en tres canales de color (RGB). La arquitectura de la red consta de varias capas convolucionales, seguidas de capas de max pooling para reducir la dimensionalidad y extraer características importantes de las imágenes. Esto se alinea con la descripción de una CNN que utiliza capas



convolucionales para extraer características y reducir la dimensionalidad de las imágenes de entrada.

Las capas convolucionales se implementan utilizando la clase Conv2D de Keras, que aplica filtros convolucionales para detectar características específicas en las imágenes. Cada capa convolucional está seguida por una función de activación ReLU para introducir no linealidad en la red y ayudar en la extracción de características más complejas.

Se utilizan capas de agrupación máxima (MaxPooling2D) para reducir el tamaño espacial de la representación de las características y hacer que la red sea más computacionalmente eficiente.

Se añaden capas de regularización en forma de dropout para evitar el sobreajuste durante el entrenamiento, lo que ayuda a mejorar la generalización del modelo.

La capa completamente conectada al final de la red utiliza activación ReLU y softmax para calcular las probabilidades de clasificación para cada clase.

Por último, el modelo se compila utilizando la función de pérdida de entropía cruzada categórica y el optimizador Nadam, tal como se describió anteriormente. Además de la precisión, también se utilizan métricas personalizadas como la puntuación F1, la matriz de confusión y las métricas de precisión y recall para evaluar el rendimiento del modelo durante el entrenamiento y la validación.

## VII. ENTRENAMIENTO DEL MODELO

El modelo se entrena en un conjunto de datos de entrenamiento a lo largo de un número específico de épocas. Durante este proceso de entrenamiento, se utilizan generadores de datos (train\_generator y valid\_generator) para suministrar lotes de datos de entrenamiento y validación, respectivamente. Además, el número de pasos por época se establece como la mitad

del tamaño de los generadores para garantizar un uso completo de los datos en cada época.

Se define un total de 20 épocas para el entrenamiento, lo cual se considera suficiente para permitir que el modelo converja y alcance un nivel deseado de precisión. La elección de este número de épocas se basa en la observación del rendimiento del modelo en el conjunto de validación durante el entrenamiento. El objetivo es alcanzar una precisión (accuracy) de al menos 0.7, lo que indica un nivel aceptable de rendimiento del modelo en la clasificación de los datos de validación.

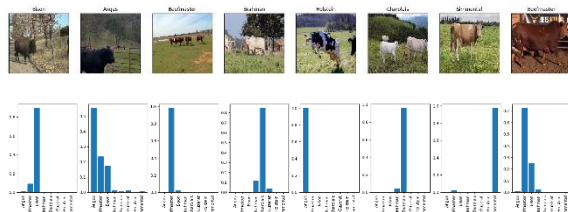
E	Pérdida	Acierto	F1	Precisión	Recall
1	1.3509	0.5080	3.2965	3250002	2.3482
2	0.6605	0.7814	1.5872	1.6295	1.5566
3	0.4271	0.8589	1.2807	1.2501	1.3179
4	0.3350	0.8900	1.1821	1.1372	1.2338
5	0.2418	0.9214	1.1028	1.0488	1.1658
6	0.2140	0.9306	1.0740	1.0182	1.1395
7	0.1747	0.9416	1.0442	0.9883	1.1103
8	0.1550	0.9483	1.0348	0.9776	1.1017
9	0.1392	0.9545	1.0123	0.9561	1.0783
10	0.1341	0.9553	1.0206	0.9627	1.0887
11	0.1241	0.923	0.9938	0.9362	1.0619
12	0.1007	0.9672	0.9922	0.9351	1.0594
13	0.1051	0.9652	0.9906	0.9331	1.0586
14	0.1036	0.9663	0.9895	0.9319	1.0576
15	0.0842	0.9739	0.9816	0.9241	1.0496
16	0.0928	0.9702	0.9830	0.9254	1.0506
17	0.0719	0.9778	0.9742	0.9170	1.0420
18	0.0628	0.9786	0.9655	0.9076	1.0339
19	0.0721	0.9748	0.9726	0.9149	1.0405
20	0.0749	0.9748	0.9736	0.9158	1.0415

**Figura 6:** resultados de entrenamiento del modelo

## VIII. VALIDACIÓN DEL ENTRENAMIENTO

Antes de presentar las métricas de validación del desempeño del modelo, es crucial exhibir las pruebas realizadas por el modelo para visualizar los resultados obtenidos. Para esto, se puede implementar un código que seleccione una

cantidad determinada de imágenes aleatorias del conjunto de datos y luego utilice el modelo de CNN para predecir la clase de cada muestra seleccionada aleatoriamente. Esto proporcionará un resultado tangible que facilita la evaluación del rendimiento del modelo en los datos de prueba.



**Figura 7:** validación del entrenamiento

Esta distribución de imágenes ilustra cómo el modelo ha sido utilizado para predecir la clase de cada muestra. En la parte superior de la figura, se presenta la imagen junto con su clase real, mientras que en la parte inferior se muestra una gráfica que representa todas las clases posibles junto con la predicción realizada por el modelo.

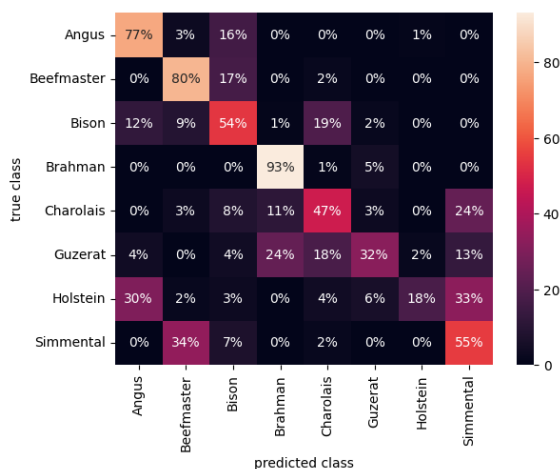
En términos generales, el modelo realiza predicciones satisfactorias para las imágenes presentadas, asignando un porcentaje a cada clase según su predicción. En algunos casos, la predicción del modelo se acerca al 100% de confianza, mientras que en otros casos, la confianza disminuye debido a características similares entre las clases. Sin embargo, en la mayoría de los casos, el modelo realiza predicciones precisas y acertadas.

## IX. VALIDACIÓN DE METRICAS

Ahora, se utilizan secciones de código que permiten mostrar gráficamente los resultados del modelo después de entrenar y validar, para esto, se utilizó la matriz de confusión y la curva ROC

### A. MATRIZ DE CONFUSIÓN

Para mostrar la matriz de confusión, se utiliza la biblioteca 'seaborn' para generar un mapa de calor. Esta representación visual facilita la comprensión de cómo el modelo clasifica las muestras en cada clase. Además, puede ayudar a identificar patrones de error, como confusiones entre clases similares. El mapa de calor muestra las frecuencias de las clasificaciones correctas e incorrectas, lo que proporciona una visión clara del desempeño del modelo en cada clase.



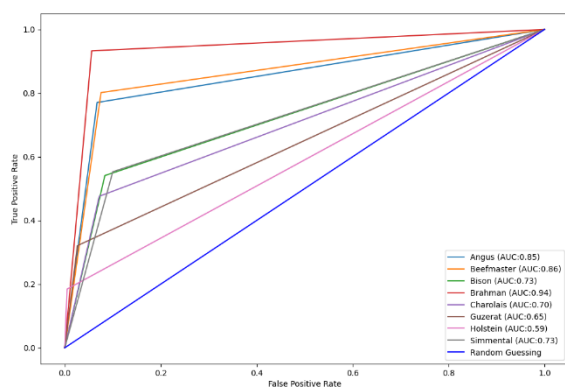
**Figura 8:** matriz de confusión del modelo

### B. CURVA ROC

En el caso del problema multiclase abordado en el proyecto, se emplea una técnica de binarización de etiquetas para calcular las curvas ROC y el área bajo la curva (AUC) para cada clase. Esta tarea se realiza utilizando la biblioteca 'scikit-learn'. La puntuación de ROC AUC es una medida que evalúa la capacidad del modelo para distinguir correctamente entre las clases. Su valor oscila entre 0 y 1, donde 1 representa un rendimiento perfecto y 0.5 equivale a un rendimiento aleatorio o al azar. Un puntaje más alto indica un mejor rendimiento del modelo.

En el caso específico de esta figura, el puntaje de ROC AUC es: 0.7558907603250962. Este resultado señala que el modelo exhibe una

capacidad destacada para diferenciar entre las clases y realizar predicciones precisas.



**Figura 9:** curva ROC del modelo

## X. ANALISIS DE RESULTADOS

El modelo de aprendizaje automático desarrollado para clasificar imágenes de razas de ganado bovino ha logrado una precisión del 76%. Este resultado indica una capacidad prometedora para reconocer y categorizar las diferentes razas de ganado bovino en las imágenes, a pesar de la presencia de algunas imágenes con baja calidad o la presencia de otros objetos que podrían dificultar la clasificación.

Sin embargo, es importante destacar que aún existe margen de mejora para alcanzar una precisión aún mayor. Se pueden explorar diversas estrategias para optimizar el modelo, como la recopilación de un conjunto de datos más amplio y diverso, el ajuste de los hiperparámetros del algoritmo o la implementación de técnicas avanzadas de preprocesamiento de imágenes. Estas estrategias podrían ayudar a abordar los desafíos y las limitaciones actuales, lo que permitiría mejorar la precisión y la robustez general del modelo.

Al analizar la matriz de confusión, se pueden identificar algunos desafíos específicos en la clasificación de ciertas razas. Por ejemplo, se observó que aproximadamente el 19% de las muestras de la clase 'Angus' fueron erróneamente clasificadas como 'Beefmaster'. Además, se encontraron dificultades en la clasificación de las razas 'Guzerat' y 'Holstein', donde el área bajo la curva (AUC) fue relativamente baja (0.65 y 0.59 respectivamente).

Estos errores de clasificación resaltan la necesidad de mejorar la capacidad del modelo para distinguir con precisión entre las razas de ganado bovino. Se podría realizar un análisis más detallado de las características distintivas y los patrones visuales presentes en las imágenes de cada raza para mejorar la precisión de la clasificación. Además, ampliar y diversificar el conjunto de datos de entrenamiento, especialmente con ejemplos representativos de las razas problemáticas, podría ayudar al modelo a mejorar su capacidad de generalización y clasificación precisa.

## XII. REFERENCIAS

1. Sarizeybek, A. T., & Isik, A. H. (2024). Detection of Bovine Species on Image Using Machine Learning Classifiers. *Gazi University Journal of Science*, 37(1), 137-148. <https://doi.org/10.35378/gujs.1203685>
2. Jogi, R., Temburnikar, G., Jadhav, A., Biradar, A., Gajbhiv, S., & Malge, A. (2024). Cattle Breed Classification Techniques: Framework and Algorithm Evaluation. *Journal of Propulsion Technology*, 1739-1749.
3. S. Manoj, R. S and K. V, "Identification of Cattle Breed using the Convolutional Neural Network," 2021 3rd



International Conference on Signal  
Processing and Communication  
(ICPSC), Coimbatore, India, 2021, pp.  
503-507, doi:  
10.1109/ICSPC51351.2021.9451706.

### **XIII. LINK DEL REPOSITORIO**

[https://github.com/vladortegab/Deep\\_Learning\\_  
Proyecto\\_Clasificador\\_Imagenes](https://github.com/vladortegab/Deep_Learning_Proyecto_Clasificador_Imagenes)