



Clasificación de imágenes del dataset FBSI por medio de redes neuronales

Classification of the FBSI dataset using neural networks

José Luis Luna Bojórquez

Universidad Autónoma Indígena de México, Los Mochis, Sinaloa, México

joselb1802m0092@uais.edu.mx

ORCID: 0009-0007-9696-7660

Alan David Ramírez Noriega

Universidad Autónoma de Sinaloa, Los Mochis, Sinaloa, México

alandramireznoriega@uas.edu.mx

ORCID: 0000-0002-8634-9988

Yobani Martínez Ramírez

Universidad Autónoma de Sinaloa, Los Mochis, Sinaloa, México

yobani@uas.edu.mx

ORCID: 0000-0002-4967-9187



<https://doi.org/10.36825/RITI.13.32.006>

Recibido: Junio 15, 2025

Aceptado: Octubre 08, 2025

Resumen: Este documento presenta la aplicación de los modelos InceptionV3, VGG16, ResNet50 y una red neuronal convolucional (CNN) simple, para analizar el *dataset* FBSI (*Feed Bunk Score Image*) compuesto por 1511 imágenes, agrupadas en seis clases. El objetivo principal es identificar el modelo con el mejor indicador de F1-Score, para la clasificación de imágenes en el contexto de la ganadería de precisión. El estudio se basa en un análisis comparativo de modelos pre-entrenados. Se describen las características del *dataset* FBSI. La configuración del equipo, el procedimiento metodológico, así como el software y lenguaje de programación necesarios, validando la hipótesis de que el sistema de clasificación de imágenes de lectura de comederos alcanza un rendimiento superior al 85%.

Palabras clave: *Lectura de Comederos, Ganadería de Precisión, Inteligencia Artificial, Dataset FBSI, InceptionV3.*

Abstract: This paper presents the application of the InceptionV3, VGG16, ResNet50 models, and a simple convolutional neural network (CNN) to analyze the FBSI (Feed Bunk Score Image) dataset composed of 1,511 images grouped into six classes. The main objective is to identify the model with the best F1 score for image classification in the context of precision livestock farming. The study is based on a comparative analysis of pre-trained models. The characteristics of the FBSI dataset are described, as well as the equipment configuration, the methodological procedure, as well as the necessary software and programming language, validating the hypothesis that the feed bunk management image classification system achieves a performance greater than 85%.

Keywords: *Precision Livestock Farming, Feed Bunk Management, Artificial Intelligence, Dataset FBSI, InceptionV3.*

1. Introducción

La engorda de ganado bovino en corrales permite controlar la alimentación y obtener una alta conversión alimenticia, es decir, un aumento de peso eficiente, la lectura de comederos es esencial para determinar la cantidad de alimento residual, evitando pérdidas económicas por desperdicio o enfermedades ocasionadas por la descomposición del alimento acumulado en el comedero [1], Antes de suministrar la ración del alimento, todos los días a la misma hora, una persona hace la lectura del comedero, mediante la observación y determina la cantidad de alimento residual de la servida anterior y la clasifica en seis categorías o clases de acuerdo al sistema de clasificación desarrollado por el Dr. Robbi Pritchard de la SDSU (*South Dakota State University*), con el fin de evaluar si la cantidad de alimento servido es el adecuado o es necesario realizar un ajuste [2], La ganadería de precisión busca optimizar estas tareas. Mediante la utilización de cámaras y técnicas de inteligencia artificial. Es posible medir la cantidad de alimento en el comedero y clasificar la cantidad de alimento residual [3], por lo que se propone la hipótesis: el sistema de clasificación de imágenes de lectura de comederos alcanza un rendimiento superior al 85%.

Primeramente, se analizaron artículos de clasificación de imágenes relacionados con frutas, porque presentan desafíos de clasificación multiclase, con variaciones de iluminación y texturas heterogéneas, haciendo énfasis en la experimentación, se identificó la metodología, las métricas y modelos utilizados. Para validar la hipótesis, el presente estudio analiza el dataset FBSI [4], utilizando la plataforma de Google Colab, para evaluar los modelos InceptionV3, VGG16, ResNet50 y Simple CNN.

2. Revisión de la literatura y trabajos relacionados

Se encontraron diversos trabajos de clasificación desde objetos y rostros hasta frutas y vegetales, sin embargo, la clasificación de comederos en corrales ha sido poco investigada encontrando el trabajo *A computer vision system for feed bunk management in beef cattle feedlot* [3], y *modelo de aprendizaje para classificação de escore de cocho em confinamentos de bovinos de corte com base em imagens* [5], estos últimos se toman como referencia y además se realizó un análisis de estudios, en el campo de la clasificación de imágenes.

La clasificación de comederos se divide en seis clases, 0, 0.5, 1, 2, 3, 4 el significado de cada clase se describe en la Tabla 1 de este trabajo en donde cada clase sugiere un ajuste determinado.

Tabla 1. Clasificación de comederos.

Clase	Descripción
0	No queda alimento en el comedero
0.5	Queda alimento disperso, la mayor parte del fondo del comedero está expuesta
1	Queda una capa fina de alimento, aprox. un grano de maíz de profundidad
2	25% a 50% de alimento restante
3	Queda más del 50% de alimento
4	El alimento está prácticamente intacto

Fuente: Sistema de clasificación desarrollado por el Dr. Robbi Pritchard de la SDSU.

La tesis de doctorado *Clasificación de la fruta del durazno mediante redes neuronales convolucionales* [6] aborda la clasificación de duraznos utilizando Redes Neuronales Convolucionales (CNN), El objetivo del experimento fue reconocer el grado de madurez de los duraznos y detectar frutos dañados, no se especificó la cantidad de imágenes del *dataset* generado, las imágenes se obtuvieron en un ambiente no controlado y las imágenes fueron recortadas para enfocarse en el área de interés, que era la fruta del durazno y se configuraron tres conjuntos de datos: Duraznos maduros e inmaduros, Duraznos sanos y dañados y Duraznos maduros e inmaduros, enfocados en características texturales. Se utilizó una Red Neuronal Convolutiva programada en Python, utilizando las librerías Keras y TensorFlow. Se realizaron pruebas con 64 imágenes, se obtuvieron las métricas de precisión,

sensibilidad y especificidad, al clasificar duraznos maduros e inmaduros, se alcanzó una precisión del 95.31%. En la clasificación de duraznos sanos y dañados, la precisión fue del 92.18%. Se observó que al clasificar más de dos grupos (maduros, inmaduros, dañados), la precisión disminuyó, lo que sugiere la necesidad de más investigaciones y el uso de redes especializadas para mejorar los resultados. Para mejorar la precisión del modelo, se sugiere que es conveniente utilizar una gran cantidad de datos, lo que implica que se deben reunir más fotografías para aumentar la confiabilidad del sistema.

El artículo *Fruit image classifier based on artificial intelligence* [7] trata sobre la implementación de un clasificador de imágenes de frutas basado en algoritmos de "Random Forest", que permita clasificar imágenes en función de los colores característicos de las frutas como parámetros únicos de identificación y clasificación. El dataset está compuesto por un total de 83 imágenes de frutas y se obtuvo a partir de la recolección de imágenes de diversas fuentes, incluyendo internet y fotografías propias. Las imágenes fueron estandarizadas en formato jpg con resolución de 500 px de ancho. Para la implementación del modelo se utilizaron las siguientes aplicaciones de software y lenguaje de programación: Python, Scikit-learn, Mahotas, OpenCV. Las métricas utilizadas para evaluar el rendimiento del modelo clasificador de imágenes de frutas fueron: Precisión, Sensibilidad, F1-Score,

El objeto del estudio en *Classification of fruits using computer vision and a multiclass support vector machine* [8], es la clasificación de frutas utilizando un método de clasificación novedoso basado en técnicas de visión por computadora y máquinas de soporte vectorial multicategoría (kSVM), para lograr una clasificación precisa y rápida de frutas a partir de un dataset de 1,653 imágenes de 18 diferentes tipos de frutas, dividido en un conjunto de entrenamiento (1,322 imágenes) y un conjunto de prueba (331 imágenes) aplicando validación cruzada estratificada de 5 pliegues, el dataset se obtuvo mediante una recolección de datos en sitio durante seis meses utilizando una cámara digital, así como la recopilación de imágenes en línea a través de un motor de búsqueda. Se implementó el modelo sin especificar software o framework utilizados y se realizaron pruebas utilizando una parte del conjunto de datos que fue dividido para las pruebas, se destaca en el artículo que la combinación de características de color, textura y forma es más efectiva que cualquier tipo de característica por sí sola, porque cada una aporta un conjunto de datos que complementa las características particulares de la fruta. Se logró la mejor precisión de clasificación del 88.2%. Las métricas utilizadas para evaluar el modelo son la Precisión y Matriz de confusión.

El objeto de estudio del artículo titulado *Classification of apples with convolutional neuronal networks* [9], es la clasificación de manzanas en categorías sanas y con daños utilizando técnicas de aprendizaje profundo, específicamente redes neuronales convolucionales (CNN), el dataset está compuesto por un total de 4800 imágenes, distribuidos en 8 clases de 600 imágenes cada una, las dimensiones de las imágenes son de 800x600 px. el dataset se obtuvo mediante la captura de imágenes de manzanas en una banda transportadora diseñada y construida para este propósito, en condiciones controladas de iluminación y un área de captura específica. Se implementaron dos modelos de redes neuronales convolucionales: LeNet5 y VGG16. Las aplicaciones de software utilizadas fueron Python 3.6, OpenCV 4.1.2, TensorFlow 2.6.0, Keras y Scikit-learn. Se realizaron múltiples pruebas, las cuales fueron analizadas con las métricas para evaluar el desempeño como la exactitud, la sensibilidad, F1Score, precisión y matriz de confusión. La conclusión del estudio es que la arquitectura LeNet5, con el tratamiento 2, fue la de mejor desempeño, logrando una exactitud del 97% en la clasificación de manzanas, lo que la convierte en la mejor opción para esta tarea en comparación con VGG16 y otros tratamientos evaluados.

El estudio *Carrot disease recognition using deep learning approach for sustainable agriculture* [10], se enfoca en la clasificación de enfermedades en la zanahoria, que ayude a identificar enfermedades de forma temprana, se utilizó un dataset de 480 imágenes, dividido en seis clases, con 80 imágenes por cada clase, una clase corresponde a zanahorias saludables y cinco a las enfermedades: Black rot, Aster yellow, Sclerotinia rot, Root knot, Growth crack. Para el desarrollo del experimento se utilizó el lenguaje de programación python, Keras y TensorFlow, aplicando técnicas de aumento de datos se alcanzaron las 10,655 imágenes, ajustadas a un tamaño de 300X300 px. se evaluaron los modelos MobileNet, VGG16, VGG19 así como el modelo InceptionV3. La comparación de los modelos se realizó utilizando las métricas de exactitud, precisión, especificidad, F1 Score y matriz de confusión, mostrando el mejor rendimiento el modelo Inception V3, con un 97.4% de exactitud.

La tesis de maestría *Modelo de aprendizaje para classificação de escurecimento de carne bovina de corte com base em imagens* [5], tiene como objetivo desarrollar una aplicación móvil para la lectura de comederos, y resolver la subjetividad de la evaluación visual actual así como la necesidad de optimizar la eficiencia alimentaria y reducir el desperdicio, propone generar una red neuronal convolucional y un dataset denominado FBSI, con imágenes de los comederos etiquetadas, fabricados con diferentes materiales, capturadas con diferentes

dispositivos, en distintas condiciones, y agrupadas en seis clases, utilizando imágenes de 108x108 píxeles el *dataset* proporciono los datos con los que fue entrenado y evaluado el modelo desarrollado, con las imágenes de prueba demostró una exactitud del 85.45%. Se encontró que Python es el lenguaje de programación más utilizado, en conjunto con Tensor Flow, Scikit-Learn y OpenCV, se aplicaron modelos de redes neuronales convolucionales y en un artículo bosques aleatorios, para evaluar el rendimiento se utilizaron las métricas como la exactitud, la precisión, sensibilidad, especificidad y F1-Score, así como la matriz de confusión.

En la Tabla 2 se muestra un comparativo de las características que se consideraron en la investigación, de los cuales se retoman las métricas utilizadas para la evaluación de los modelos y principalmente el modelo de Inception V3 para el desarrollo del experimento.

Tabla 2. Comparativa de características por artículo.

TRABAJO RELACIONADO	DATASET	MODELO	SOFTWARE	METRICAS	RENDIMIENTO
Clasificación de la fruta del Durazno mediante Redes Neuronales Convolucionales [6]	No especificado el tamaño Pruebas con 64 imágenes	CNN	Python Keras TensorFlow	Precisión Sensibilidad Especificidad	92.18%
Fruit Image Classifier Based on Artificial Intelligence [7]	83 imágenes de frutas 4 clases	Random Forest	Python Scikit-Learn Mahotas OpenCV	Precisión Sensibilidad F1 Score	No reportado
Classification of Fruits Using Computer Vision and a Multiclass Support Vector Machine [8]	1653 imágenes 18 clases	kSVM máquina soporte vectorial	No reportado	Precisión Matriz de confusión	88.2%
Classification of Apples with Convolutional Neuronal Networks [9]	4800 imágenes en 8 clases	LetNet5 VGG16	Python OpenCV TensorFlow Keras SciKit-Learn	Exactitud Sensibilidad F1 Score Precisión Matriz de confusión	97%
Carrot Disease Recognition using Deep Learning Approach for Sustainable Agriculture [10]	480 imágenes 6 clases 300x300	VGG16 VGG19 InceptionV3 MobileNet	Python TensorFlow Keras	Exactitud Sensibilidad Precisión Especificidad F1 Score Matriz de confusión	97.4%
Modelo de aprendizaje para classificação de escore de cocho em confinamentos de bovinos de corte com base em imagens [5]	Generar dataset de 1001 imágenes 6 clases	CNN	Aplicación móvil	Exactitud Matriz de confusión	85.45%

Fuente: Elaboración propia.

Aun cuando LetNet5 demostró un rendimiento del 97% al ser una de las redes neuronales convolucionales pioneras y fundamentales en el reconocimiento de dígitos manuscritos [11] presenta la limitante que el máximo tamaño de la imagen de entrada es de 32x32 px, quedando en desventaja frente a InceptionV3 con un valor típico para InceptionV3 de 299x299 px, esto significa que puede procesar imágenes más grandes, por lo tanto tiene mayor capacidad para generalizar detalles más finos, ya que al redimensionar las imágenes puede degradar la calidad [12], por lo que en la revisión de la literatura se encontró que de los modelos analizados Inception V3 es el que presenta mejor rendimiento para la clasificación de las imágenes de comederos.

3. Materiales y métodos

Debido a que se cuenta con el dataset FBSI se omitirá el proceso de adquisición de imágenes el cual es un proceso importante ya que de la calidad de los datos dependerá también la capacidad del modelo para clasificar correctamente las imágenes.

3.1. Diseño del experimento

La clasificación de imágenes por medio de sistemas de inteligencia artificial implica analizar una imagen y regresar una etiqueta con la clasificación del contenido, Este proceso se basa en modelos de Deep learning que son entrenados a partir de grandes volúmenes de datos etiquetados, permitiendo que el sistema reconozca patrones complejos en las imágenes. De los trabajos relacionados se obtiene resultados experimentales, donde se aplicaron diferentes modelos de inteligencia artificial y cuáles fueron las métricas que permitieron evaluar cada modelo.

3.2. Descripción del dataset

El *dataset Feed Bunk Score Images* (FBSI) Imágenes de lectura de comederos en corrales de engorda [4], está formado por imágenes de alimento en comederos de corrales de engorda de ganado bovino, cada imagen esta etiquetada de acuerdo con el sistema de clasificación de la universidad de Dakota del Sur (*South Dakota State University*) [2] con el objetivo de aplicar algoritmos de clasificación y generar un modelo que mejore la eficiencia en la alimentación del ganado. La Tabla 3 muestra las especificaciones del *dataset*. Se utilizó la plataforma de Google Colab, en conjunto con una carpeta en Google drive para almacenar y desde ahí cargar el *dataset* en el entorno de ejecución, seleccionando el hardware gratuito y haciendo uso de CPU,

Tabla 3. Descripción del *dataset* FBSI (*Feed Bun Score Image*).

Clasificación	No. imágenes	%	Formato de imagen	JPG
0	85	5.63	Equipo de captura	Iphone 11 y Sony DSCH300
½	266	17.60	Hora de captura	Entre 6:00 am y 10:00 am
1	265	17.54	Formato de Etiquetas	XML y CSV
2	118	7.81	Iluminación	Natural
3	270	17.87	Angulo de captura	Diferentes
4	507	33.55	Ubicación	Minas Gerais, Brasil
Total de Img.	1511	100%	Granja	Nueve diferentes

Fuente: El *dataset Feed Bunk Score Images* (FBSI) [5].

3.3. Procedimiento

Se diseñó el procedimiento para asegurar la reproducibilidad del experimento. a continuación, se describen los pasos seguidos.

3.3.1. Diseño conceptual

Es la primera actividad y consiste en definir el proceso desde la recolección de las imágenes, la preparación de los datos, construcción del modelo, el entrenamiento, hasta la evaluación, y los ajustes.

3.3.2. Recolección de imágenes

La recolección de las imágenes, no se aplica en este experimento, pero se considera como parte de un procedimiento estándar. El cual consiste en aplicar la metodología definida en el diseño conceptual donde las imágenes deben cumplir con los criterios definidos, entre ellos el etiquetado, el formato de las imágenes y todas las especificaciones que permitan obtener las características importantes, considerando la resolución adecuada, la distancia del objetivo, la iluminación, color, cantidad de imágenes, ángulos de las tomas, sombras, horarios y orientación. Dependiendo del dispositivo de captura y si la imagen no ha sido manipulada es posible que conserve los metadatos de la imagen los cuales incluyen, coordenadas gps, fecha, hora tipo de lente utilizado, los cuales enriquecen el proceso de etiquetado y el análisis posterior.

3.3.3. Preparación de los datos

La preparación de los datos, no se aplica en este experimento, pero se considera como parte de un procedimiento estándar. En esta etapa se realiza el etiquetado de las imágenes y consiste en cargar las imágenes en una aplicación “Label Estudio” y una por una se van analizando, de forma que se etiqueta en cada imagen el área de interés y se genera el archivo de las etiquetas. Las imágenes se agrupan en carpetas de acuerdo con el etiquetado y su clasificación, incluye también, sin ser estrictamente necesario la normalización del tamaño y la resolución, así como la separación en grupos de entrenamiento y prueba. Es importante señalar que el *dataset* contiene 6 clases definidas como 0, ½, 1, 2, 3 y 4 y para efectos de este estudio se renombro la clase ½ como clase 5.

3.3.4. Construcción del modelo

De acuerdo con la clasificación de las imágenes y el número de clases generadas, se construye un modelo con el *framework* de TensorFlow y dependerá de las bibliotecas utilizadas, la estructura final.

3.3.5. Entrenamiento del modelo

En la etapa de entrenamiento, ya se debe tener dividido el *dataset*, y se realizan los primeros ajustes al modelo, se debe tener cuidado de no producir un sobreajuste (*overfitting*) debido a un ajuste excesivo en los datos de entrenamiento, aplicando técnicas de aumento de datos (*data augmentation*) que consiste en aumentar los datos de entrenamiento modificando imágenes del mismo dataset mediante cambios de escala, rotaciones y recortes, esto aplica cuando la cantidad de datos de entrenamiento no es suficiente para alcanzar las métricas aceptables del modelo. Otros valores que también es importante ajustar en el entrenamiento es la tasa de aprendizaje, el tamaño del lote y el número de épocas, hasta optimizar el rendimiento del modelo, considerando realizar pruebas con modelos pre-entrenados (*Transfer Learning*). Se considera desde el tiempo de entrenamiento y velocidad para clasificar en el costo computacional, así como las métricas de Exactitud, Precisión, F1Score, aplicables para modelos de clasificación.

3.3.6. Evaluación y ajustes

Para la evaluación del modelo se utilizan datos que el sistema nunca ha visto, haciendo una evaluación imparcial, se utilizan métricas como la Precisión, Sensibilidad, F1-score y Matriz de confusión, los errores encontrados se analizan detalladamente para conocer porque fallo y se realizan los ajustes necesarios hasta alcanzar el máximo de efectividad del sistema.

3.4. Hipótesis

- **Hipótesis Nula H0**

El rendimiento del sistema de clasificación de imágenes de lectura de comederos no será mayor al 85%.

- **Hipótesis Alterna H1**

El rendimiento del sistema de clasificación de imágenes de lectura de comederos será mayor al 85%

3.5. Definición conceptual de las variables

- **Rendimiento del Sistema de Clasificación:** Se refiere a la capacidad del modelo para clasificar correctamente las imágenes, es decir del total de las clasificaciones cuantas fueron clasificadas correctamente.
- **Sistema de Clasificación de Imágenes:** Modelos de inteligencia artificial entrenados con un conjunto de datos, para que reconozcan patrones en imágenes desconocidas.
- **Lectura de Comederos:** Proceso mediante el cual se mide la cantidad de alimento que existe en el comedero
- **Rendimiento del Sistema de Clasificación:** Se medirá utilizando el F1-Score como métrica.

3.6. Métricas

Las métricas calculadas a partir de una matriz de confusión son esenciales para evaluar el rendimiento del modelo. La precisión indica cuán confiables son las predicciones positivas; la sensibilidad ó *recall* mide la capacidad del modelo para identificar correctamente todas las instancias positivas; y el F1 Score proporciona un balance entre ambas métricas. Estos resultados permiten identificar áreas donde el modelo puede ser mejorado y ayudan a entender su desempeño en cada clase.

Exactitud (Accuracy): Es el porcentaje total de predicciones correctas realizadas por el modelo entre el número total de predicciones. Se calcula con la Ecuación 1.

$$\text{Exactitud} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

donde, TP son los verdaderos positivos, TN los verdaderos negativos, FP los falsos positivos y FN los falsos negativos. Es útil en conjuntos de datos balanceados, pero puede ser engañosa en conjuntos desbalanceados.

Precisión (Precision): Mide la proporción de verdaderos positivos sobre el total de predicciones positivas realizadas por el modelo. Se calcula con la Ecuación 2.

$$\text{Precisión} = \frac{TP}{TP+FP} \quad (2)$$

es especialmente importante en contextos donde los falsos positivos tienen un costo alto, ya que indica cuán confiables son las predicciones positivas.

Sensibilidad o Exhaustividad (Recall): Indica la proporción de verdaderos positivos sobre el total de casos positivos reales. Se calcula con la Ecuación 3.

$$\text{Sensibilidad} = \frac{TP}{TP+FN} \quad (3)$$

es importante cuando los falsos negativos son costosos o peligrosos, ya que mide la capacidad del modelo para identificar correctamente todas las instancias positivas.

F1 Score: Es la media armónica entre precisión y sensibilidad, proporcionando un balance entre ambas métricas. Se calcula con la Ecuación 4.

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Sensibilidad}}{\text{Precision} + \text{Sensibilidad}} \quad (4)$$

Es útil en situaciones donde hay un desbalance entre las clases y se necesita una métrica única que considere tanto la precisión como la sensibilidad.

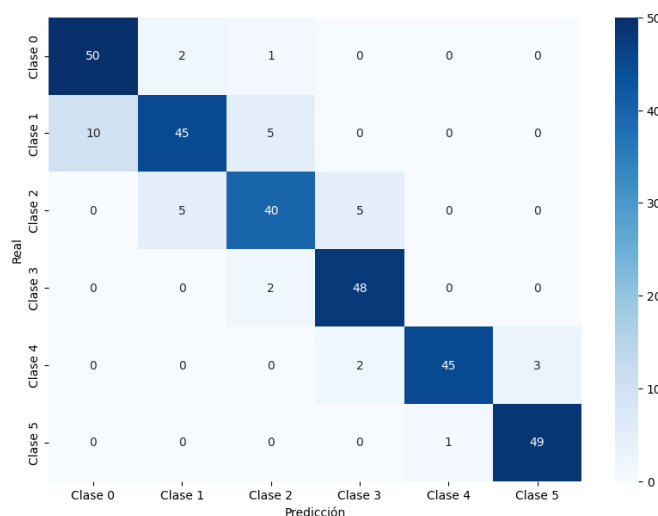


Figura 1. Matriz de confusión, seis clases. Fuente: *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent* [13].

Matriz de Confusión: Es una tabla que se utiliza para describir el rendimiento del modelo al mostrar el número de predicciones correctas e incorrectas desglosadas por clase. Cada fila representa instancias en una clase predicha mientras que cada columna representa instancias en una clase real. Permite visualizar no solo la exactitud global, sino también cómo se desempeña el modelo en cada clase específica, ayudando a identificar clases que pueden estar siendo mal clasificadas, la Figura 1 muestra una matriz de confusión donde se puede apreciar una diagonal en color azul más intenso indicando el cruce de los valores reales con los valores pronosticados. Filas: Representan las clases verdaderas, Columnas: Representan las clases predichas, Valores: Indican el número de instancias clasificadas en cada combinación de clase verdadera y clase predicha. La diagonal principal muestra las predicciones correctas, Los valores fuera de la diagonal indican errores de clasificación.

4. Resultados

La ganadería de precisión es un área de la tecnología que está en crecimiento y con la integración de la inteligencia artificial, permite realizar análisis de imágenes de lectura de comederos, optimizando los sistemas de producción.

Este estudio propone un sistema inteligente para la lectura de comederos en corrales de engorda de ganado bovino, para sustentarlo se estudiaron y compararon los modelos de clasificación de imágenes de los trabajos relacionados, además en base a una revisión de la literatura, que incluyó comparaciones de rendimiento de modelos InceptionV3, VGG16, ResNet-152 y Xception [14], se determinó evaluar los modelos: InceptionV3, VGG-16, ResNet-50 y Simple CNN.

Utilizando Python y la plataforma Google Colab, se configuró: epochs=20, batch size=32, imagen size=128x128, el proceso metodológico incluyó la carga del dataset FBSI, el preprocesamiento, la creación de los modelos, el aumento de datos y la evaluación de los modelos, Los resultados indican que el modelo InceptionV3 fue el más efectivo. En la Tabla 4 se muestra una comparativa de las métricas de cada modelo.

Tabla 4. Comparación del rendimiento de los modelos con los datos de prueba.

Modelo	Exactitud	Precision (Avg)	Sensibilidad (Avg)	F1-Score (Avg)
Simple CNN	0.528053	0.447438	0.413899	0.390137
ResNet50	0.333333	0.138704	0.169492	0.088590
VGG16	0.660066	0.659696	0.566527	0.577015
InceptionV3	0.679868	0.648910	0.613966	0.600424

Fuente: Elaboración propia.

Las Figuras de la 2 a la 5 muestran las gráficas de entrenamiento y validación para la exactitud (*accuracy*) y la pérdida, gráficamente se puede apreciar cuando un modelo experimentó un sobre ajuste o sub ajuste, mostrando el punto en el que el entrenamiento se detuvo debido a la opción de *early stopping* para prevenir el *overfitting*.

La Figura 2 muestra un rendimiento promedio de *F1-Score* de 0.390137 para el modelo Simple CNN el cual tiene una exactitud de 0.528053.

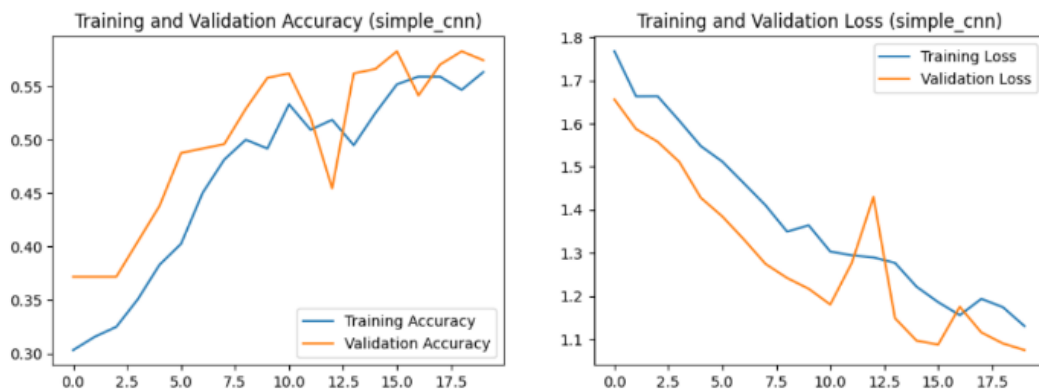


Figura 2. Rendimiento del Modelo Simple CNN. Fuente: Elaboración propia.

El entrenamiento y validación para el modelo VGG16 se presenta en la Figura 3 mostrando un rendimiento promedio de *F1-Score* de 0.577015 y una exactitud de 0.660066.

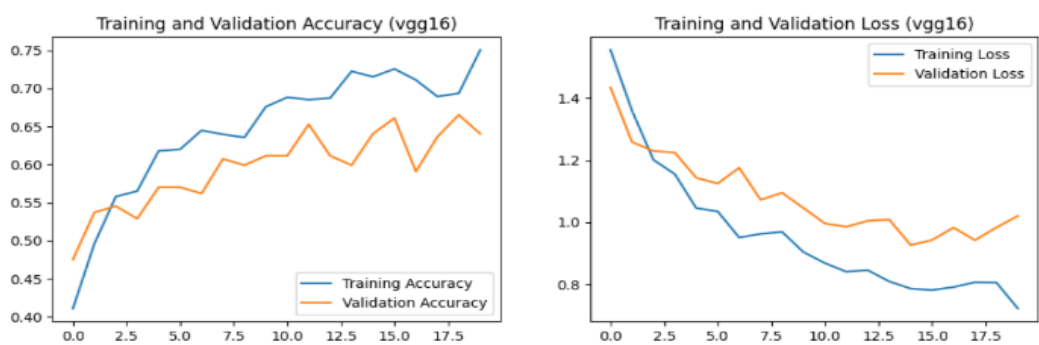


Figura 3. Rendimiento del modelo VGG16. Fuente: Elaboración propia.

El rendimiento promedio *F1-Score* del modelo ResNet50 es de 0.088590 se muestra en la Figura 4, con una exactitud de 0.333333 el entrenamiento se detuvo en la época nueve, debido a la activación de “*Early Stopping*” una técnica implementada, que cuando detecta que una métrica deja de mejorar detiene el entrenamiento, lo que ayuda a prevenir el sobre ajuste y el ahorro de tiempo.



Figura 4. Rendimiento del Modelo ResNet50. Fuente: Elaboración propia.

El modelo InceptionV3 revela un valor promedio de *F1-Score* de 0.600424 por arriba del modelo VGG16 que mostro un valor de 0.577015, esa diferencia, lo sitúa como el modelo con mejor rendimiento del sistema de clasificación de imágenes de lectura de comederos. La Figura 5 muestra el progreso del entrenamiento y validación de la exactitud y la pérdida.

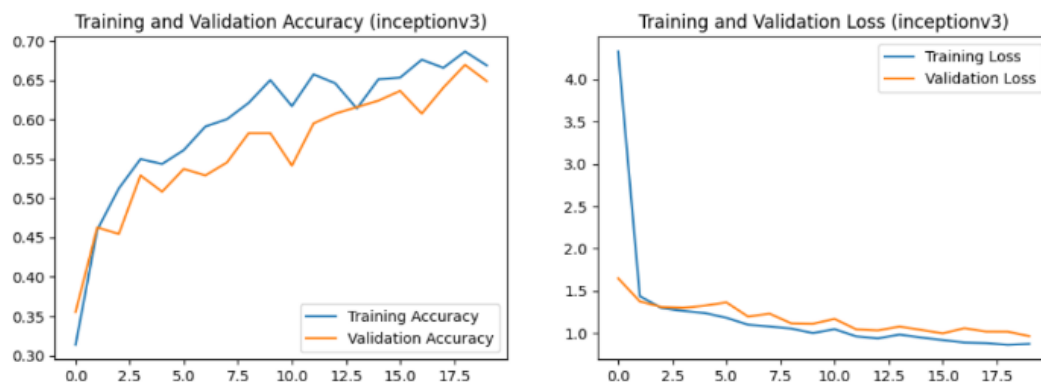


Figura 5. Rendimiento del Modelo InceptionV3. Fuente: Elaboración propia.

En las Figuras de la 6 a la 10 se muestran las matrices de confusión, generadas con los datos de prueba de cada modelo, se presenta una vista detallada del rendimiento de clasificación por cada clase, indicando el número de instancias en la que cada clase fue clasificada correctamente en línea diagonal y los valores con errores hacia arriba o debajo de la diagonal, permitiendo distinguir las clases en las que se confunde el modelo y con qué frecuencia.

La Figura 6 muestra la matriz de confusión generada del modelo Simple CNN, donde se puede apreciar errores de clasificación en todas las clases, lo que confirma los valores bajos de *F1-Score* presentados en la Tabla 4.

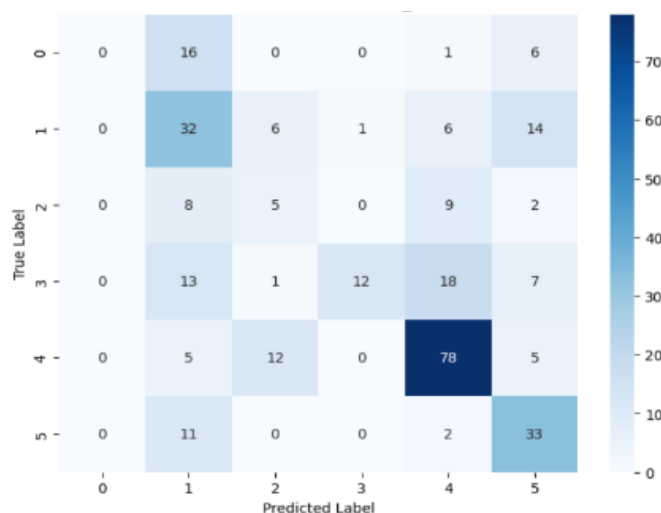


Figura 6. Matriz de confusion modelo Simple CNN. Fuente: Elaboración propia.

La Figura 7 corresponde a la matriz de confusión del modelo ResNet50, se puede apreciar que asigno todas las clasificaciones a la clase 4, confirmando por qué los valores más bajos de *F1-Score* de los cuatro modelos presentados en la Tabla 4.

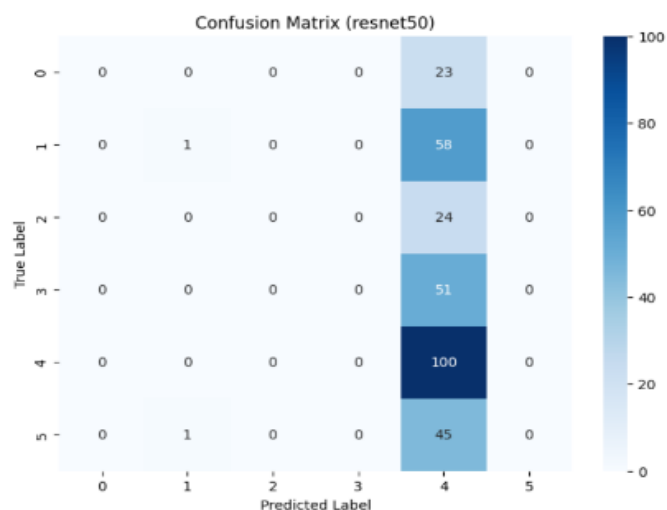


Figura 7. Matriz de confusión del modelo ResNet 50. Fuente: Elaboración propia.

El modelo VGG16 en la matriz de confusión de la Figura 8 muestra errores dispersos en todas las clases, su rendimiento quedó en segundo lugar después del modelo InceptionV3.

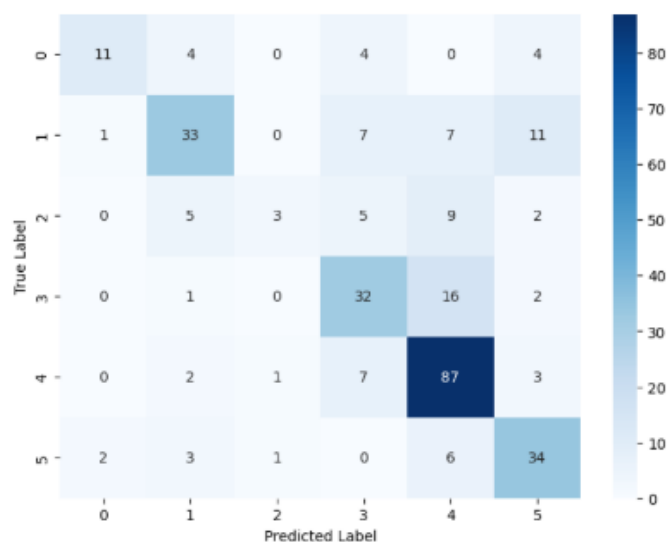


Figura 8. Matriz de confusión del modelo VGG16. Fuente: Elaboración propia.

La Figura 9 muestra la matriz de confusión del modelo InceptionV3, donde se puede apreciar que la mayoría de los errores lo comete en la clase 1, en menor medida la clase 2 y menos aún en la clase 3 y de acuerdo con los datos mostrados en la Tabla 4, la Figura 5 y la Figura 9, el modelo InceptionV3 mostró los mejores rendimientos al evaluar el *dataset* FBSI.

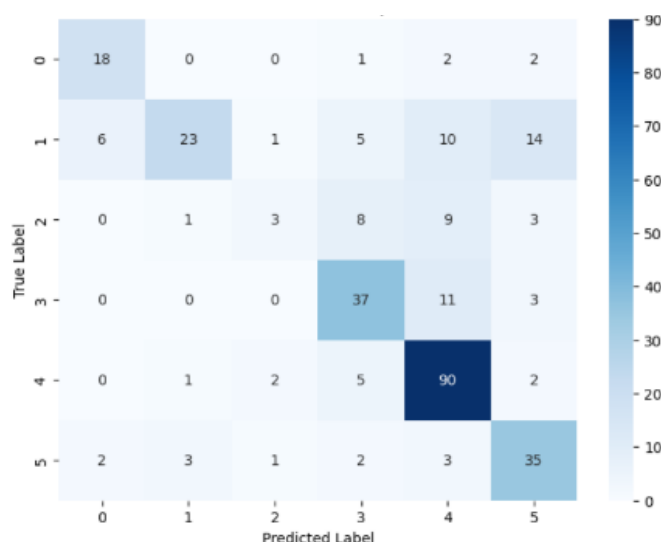


Figura 9. Matriz de confusión del modelo Inception V3. Fuente: Elaboración propia.

Al central el experimento en el modelo de InceptionV3, y realizar ajustes en el tamaño de las imágenes de entrada para el que fue diseñado InceptionV3 que es de 299x299px, *epochs*=30, *batch size*=32, descongelando las últimas 100 capas se alcanzó una notable mejoría, llegando al 92% de rendimiento como se puede observar en la Figura 10, superando al modelo desarrollado por Marques de Paula donde alcanza un rendimiento del 85% en el *dataset* FBSI.

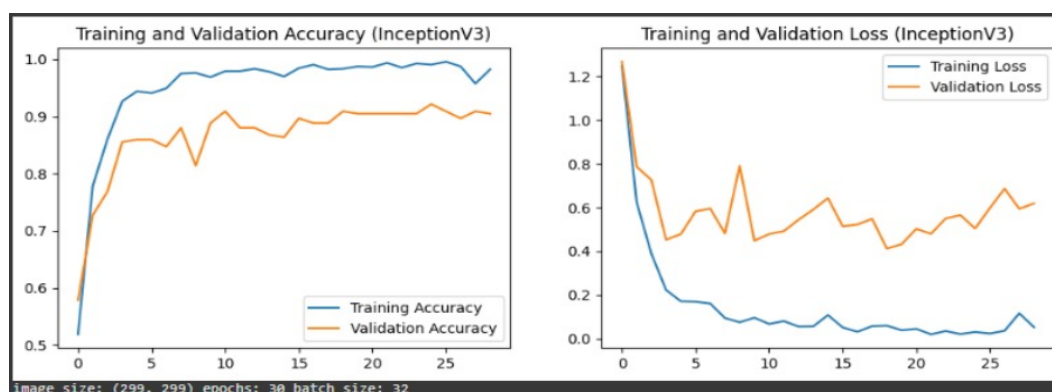


Figura 10. Entrenamiento exactitud y pérdida del modelo InceptionV3 con ajustes básicos en los hiperparámetros. Fuente: Elaboración propia.

En la matriz de confusión de la Figura 9, se exhibe una dispersión de valores fuera de la diagonal principal, indicando errores de clasificación, y después de los ajustes en los hiper parámetros, la Figura 11 muestra la matriz de confusión, una concentración significativamente mayor de valores correctos a lo largo de la diagonal principal, lo cual demuestra una mejora del desempeño del modelo.

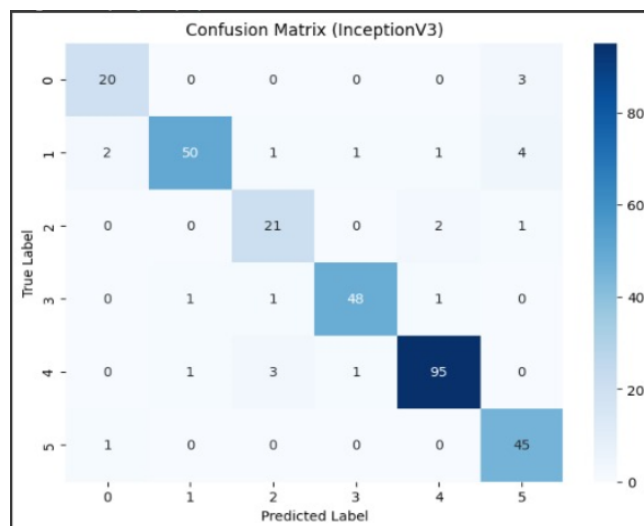


Figura 11. Matriz de confusión con ajustes en hiperparámetros. Fuente: Elaboración propia.

Al realizar un análisis profundo del rendimiento del modelo InceptionV3, como se muestra en la tabla 5 es importante indicar que las clases 0 y 2 contienen el menor número de imágenes del *dataset* con el 5.63% y 7.81% respectivamente y el rendimiento del modelo fue de los más bajos por clase con 0.87 y 0.84 de F1-Score.

Tabla 5. Resultados por clase del modelo inceptionV3.

Clase	Precision	Sensibilidad	<i>F1-Score</i>	<i>Support</i>
0	0.870	0.870	0.870	23
1	0.962	0.847	0.901	59
2	0.808	0.875	0.840	24
3	0.960	0.941	0.950	51
4	0.960	0.950	0.955	100
5	0.849	0.978	0.909	46
<i>Accuracy</i>				0.921
<i>Macro Avg</i>	0.901	0.910	0.904	303
<i>Weighted Avg</i>	0.924	0.921	0.921	303

Fuente: Elaboración propia.

Accuracy (Exactitud): El modelo muestra una exactitud general del 92.08%, indicando que el modelo es efectivo en general. El rendimiento en la clase 0 para la precisión, Sensibilidad y *F1-score* son del 87%, la clase 1 tiene una precisión de 96.2% y un Sensibilidad del 84.7%, resultando en un *F1-score* de 90.1%. la clase 2 muestra un rendimiento más bajo, con una precisión del 80.8%, un Sensibilidad del 87.5% y *F1-score* del 84.0%, Para la clase 3 muestra un rendimiento del *F1-score* de 95.0% precisión del 96.0% y Sensibilidad del 94.1%, la clase 4, presenta un rendimiento con un *F1-score* de 95.5% precisión del 96.0% y Sensibilidad del 95.0%, La clase 5 para la Sensibilidad un 97.8%, significa que el modelo es muy bueno para detectar la mayoría de todas las de esta clase, la precisión del 84.9%, resultando un *F1-score* de 90.9%.

5. Conclusiones

En esta investigación se evaluó una red neuronal convolucional simple y tres arquitecturas pre-entrenadas, ResNet50, VGG16 e InceptionV3, para la clasificación de imágenes de lectura de comederos, la metodología

incluyo técnicas de aumento de datos y de control como “*Early Stopping*”, los reportes de las métricas obtenidas y las matrices de confusión demuestran que los modelos pre-entrenados superan a la red neuronal convolucional simple.

Específicamente el modelo InceptionV3, presentó el mejor rendimiento, logrando superar la hipótesis establecida, como lo demuestra la investigación de reconocimiento de enfermedades de la zanahoria [10], las gráficas de entrenamiento y validación (Figuras 2-5) demuestran como el modelo InceptionV3 representa una mayor estabilidad durante el entrenamiento y se refleja como lo muestran las matrices de confusión (figuras 6-9), a diferencia de los modelos Simple CNN y ResNet50 que generaron el mayor número de clasificaciones erróneas. el modelo VGG16 quedo en segunda posición.

Como trabajos futuros se considera ampliar el conjunto de datos, con la intención de balancear la distribución de las clases, para evaluar nuevamente el rendimiento del modelo con un *dataset* más robusto y realizar un reajuste de hiper parámetros. Además, validar su aplicabilidad en escenarios reales de ganadería de precisión.

6. Referencias

- [1] Ceconi, I., Ferreyra, S., Méndez, D., Davies, P. (2018). Terminación a corral: lectura de comederos y su impacto sobre resultados productivos. *Revista Ganado*, 50-54. https://primianutricion.com.ar/wp-content/uploads/2019/08/inta_dg2018_ceconi_terminacion_corral_lectura.pdf
- [2] Rusche, W. (2023). *Feed Bunk Management*. <https://extension.sdstate.edu/feed-bunk-management>
- [3] Dorea, J. R. R., Cheong, S. (2019). PSXI-2 A computer vision system for feed bunk management in beef cattle feedlot. *Journal of Animal Science*, 97(Supplement 3), 389–390. <https://doi.org/10.1093/jas/skz258.776>
- [4] Marques de Paula, B., Rezenda da Silva, G., Ferreira, S. E., Coimbra Maia, B. L., Gomes Almeida, M. E., Soares Júnior, V. M., da Silva Maciel, L. M., Chaves, A. S. (2023). Dataset of feed bunk score images of cattle feedlot. *Data in Brief*, 47, 1-6. <https://doi.org/10.1016/j.dib.2023.108996>
- [5] Marques De Paula, B. (2023). *Modelo de aprendizaje para classificação de escore de cocho em confinamentos de bovinos de corte com base em imagens* [Tesis de Maestría]. Universidade Federal De Minas, Montes Claros, Brasil. <https://hdl.handle.net/1843/69269>
- [6] Arévalo Zenteno, M. D. (2021). *Clasificación de la fruta del Durazno mediante Redes Neuronales Convolucionales* [Tesis de Doctorado]. Universidad Autónoma del Estado de México. Texcoco, Estado de México.
- [7] Heras, D. (2017). Fruit image classifier based on artificial intelligence. *Revista Killkana Técnica*, 1 (2), 21–30. https://killkana.ucacue.edu.ec/index.php/killkana_tecnico/article/view/79/127
- [8] Zhang, Y., Wu, L. (2012). Classification of fruits using computer vision and a multiclass support vector machine. *Sensors*, 12 (9), 12489–12505. <https://doi.org/10.3390/s120912489>
- [9] Olguín-Rojas, J. C., Vasquez-Gomez, J. I., López-Canteñs, G. de J., Herrera-Lozada, J. C. (2022). Classification of apples with convolutional neuronal networks. *Revista Fitotecnica Mexicana*, 45 (3), 369–378. <https://doi.org/10.35196/rfm.2022.3.369>
- [10] Rashid Methun, N., Yasmin, R., Begum, N., Rajbongshi, A., Islam, E. (2021). Carrot Disease Recognition using Deep Learning Approach for Sustainable Agriculture. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 12 (9), 723-741. <https://dx.doi.org/10.14569/IJACSA.2021.0120981>
- [11] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86 (11), 2278-2324. <https://doi.org/10.1109/5.726791>
- [12] Jerubbaal, J. L., Rajkumar, J., Mahesh, B. (2019). Impact of image size on accuracy and generalization of convolutional neural networks. *International Journal of Research and Analytical Reviews*, 6 (1), 70- 80. <https://n9.cl/sqkuc>
- [13] Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Inc. <https://n9.cl/kntzd>
- [14] Chollet, F. (2017). *Xception: Deep Learning with Depthwise Separable Convolutions*. Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA. <https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.195>