

Phenotypic Evaluation of Brown Swiss Dairy Cattle Using Images Processing

Hugo Alatrística-Salas, *Member, IEEE*, Julianna Milagros Apumayta Lopez, Eduardo Leuman Fuentes Navarro, and Miguel Nunez-del-Prado, *Member, IEEE*,

Abstract—Phenotypic evaluation of Brown Swiss cows is a method used in the Peruvian Andean Region to identify and select breeding females. Selection is based on their closeness to ideal dairy conformation. This task is performed by specialists in stock judging. Under this context, the aim of the present study was to demonstrate the feasibility to perform a partial phenotypic evaluation of Brown Swiss cows by overlapping templates through development of a cow detection model and a decision making support system for identification and automatic classification of Brown Swiss cattle. TensorFlow Object Detection API was used to detect the cow in real time. The learning transfer approach was used for training, and MobilNet was selected as a pre-trained architecture. As result a mobile app was developed to determine whether an animal has Brown Swiss breed phenotypic characteristics through an automatic adjustment and calibration of a cow's template.

Index Terms—Object detection, brown swiss, TensorFlow, MobileNet, phenotypic.

I. INTRODUCCIÓN

Brown Swiss es la raza de ganado vacuno más extendida en el Perú. Esta raza puede encontrarse en zonas que se encuentran ubicadas al nivel del mar hasta en zonas que bordean los 4000 m.s.n.m. En la región Andina, su presencia es importante; estando en número sólo por debajo del ganado criollo de origen español y sus cruces respectivos [1]. La mayoría de los productores de esta región realiza la selección genética de sus animales a partir de evaluaciones lineales de fenotipo (características morfológicas del animal), en contraposición a las evaluaciones genómicas, las cuales son cada vez más usadas a nivel mundial. Dicha práctica es muy arraigada en toda Latinoamérica [2], estando avalada por diversos estudios que reportan una correlación existente entre características de tipo y la producción de leche [3]. Las evaluaciones lineales de fenotipo son empleadas también en programas de cruzamientos, donde luego de identificar las características de tipo que las vacas necesitan mejorar, se programan apareamientos con toros que pueden aportar a una mejor descendencia en las características deseadas.

En la región Andina del país, la Asociación Brown Swiss del Perú, entrega certificados de registro de pedigrí a productores que deseen contar con información oficial de la descendencia de sus animales Brown Swiss; aplicando para la incorporación de nuevos ejemplares una evaluación lineal que comprende la valoración imparcial de la conformación del ganado lechero

comparándolo con un tipo ideal, teniendo el animal que alcanzar un puntaje mínimo para ser considerado de la raza [4]. De esta manera, se garantiza la pureza racial de los animales y se contribuye al mejoramiento genético de los mismos.

Si bien la evaluación lineal es requisito para mantener los estándares fenotípicos de la raza Brown Swiss en zonas altoandinas del Perú, la realización de esta evaluación, así como el registro genealógico de animales es limitado por la escasa disponibilidad en el país de personal altamente calificado para la realización de esta prueba.

Con el fin de apoyar la identificación y clasificación de ganado se han realizado diferentes investigaciones utilizando algoritmos de visión computacional, técnicas de *Machine Learning* y algoritmos de reconocimiento de patrones [5]. Estas investigaciones se enfocan en identificar el ganado vacuno a través de la extracción de características biométricas [5], o reconocimiento del comportamiento del ganado vacuno [6]. Además, se han desarrollado sistemas basados en zoometría animal, con la finalidad de apoyar el proceso de medición y evaluación a fin de determinar cual de estos mejora genéticamente las debilidades presentes en la generación actual del vacuno [7].

En ese sentido, el presente trabajo describe el desarrollo de una aplicación móvil que automatiza el proceso general de evaluación de las características fenotípicas del ganado vacuno de la raza Brown Swiss. Para ello, la aplicación móvil propuesta posee un sistema de sincronización en ausencia de acceso a Internet; es decir, la aplicación funciona en modo *off-line* y cuando la conexión se recupera, esta se sincroniza y funciona en modo *on-line*. La solución propuesta permitirá una evaluación fenotípica del ganado vacuno más precisa que la evaluación visual, de modo tal que los ganaderos puedan tener una cuantificación del estado del animal *in-situ* y en tiempo real.

El presente trabajo está organizado de la siguiente forma. La Sección II aborda el estado del arte. Luego, la Sección III describe el juego de datos usado, mientras que la Sección IV detalla la metodología y las experimentaciones realizadas sobre la base de datos real, mientras que en la Sección V se muestran y discuten los resultados. Finalmente, la Sección VI presenta las conclusiones y trabajos futuros.

II. ESTADO DEL ARTE

En los últimos años, se han realizado diferentes investigaciones sobre la identificación y clasificación de ganado para mejorar su producción y pureza racial, entre las que resaltan los estudios basados en biometría y zoometría animal.

Hugo Alatrística-Salas and Miguel Nunez-del-Prado are with Universidad del Pacífico. Eduardo Leuman Fuentes Navarro is with Universidad Nacional Agraria la Molina and Julianna Milagros Apumayta Lopez is with Pontificia Universidad Católica del Perú.

La Biometría animal es un campo de investigación emergente que utiliza visión computacional y reconocimiento de patrones para realizar la extracción de características. La idea detrás de sistemas de reconocimiento basados en biometría animal es extraer características discriminantes para identificar la apariencia fenotípica de especies o analizar el comportamiento de animales en función de un patrón morfológico y características biométricas [8].

Entre los sistemas basados en Biometría animal, podemos mencionar dos trabajos importantes. Por un lado, el sistema de identificación automática para ganado utilizando extracción de características híbridas y paradigmas de clasificación sobre imágenes del hocico del animal, desarrollado en [5]. En esta técnica, se analizaron 500 imágenes de ganado con 10 imágenes del hocico por animal, haciendo un total de 5000 imágenes. Estas imágenes fueron segmentadas en 3 clases: clase A de 2000 imágenes de hocico con iluminación, clase B de 2000 imágenes de hocico borrosas y con iluminación y la clase C de 1000 imágenes de hocico borrosas, con oclusión e iluminación. Luego, se analizó específicamente el hocico de los animales, asociándolas como huella única, ya que a partir de los hocicos se pueden extraer características distintivas entre cada animal. Se utilizaron técnicas de extracción de características de textura como Haralick, entre otros modelos basados en técnicas de apariencia, como *Histogram of Oriented Gradients* (HOG) [9], características de *Wavelet*, características de color, algoritmo de Tamura, algoritmo de *Speeded Up Robust Features* (SURF) [10], *Local Binary Patterns* (LBP) [11] y *Fuzzy Local Binary Patterns* (Fuzzy-LBP). Para realizar la clasificación, los autores emplearon diferentes algoritmos de clasificación como el *K-Nearest Neighbors* (KNN) [12], *fuzzy KNN* [12], *Radial Basis Perceptron Networks* (RBPN) [13], *Decision Trees* (DT) [14], *Gaussian Mixture Models* (GMM) [15], *Artificial Neural Networks* [16], *Probabilistic Neural Networks* (PNN) [17], *Multi Layer Perceptrons* (MLP) [18] y *Naive Bayes* [19]. Como resultado, el clasificador K-NN obtuvo la mejor precisión al 96.74 % para la clase A (*i.e.*, 2000 imágenes de hocico con iluminación), 96.34 % para la clase B (*i.e.*, 2000 imágenes de hocico borrosas y con iluminación) y 89.95 % para la clase C (*i.e.*, 1000 imágenes de hocico borrosas), al clasificar y reconocer al ganado individual. Posteriormente, los mismos autores mejoraron sus resultados anteriores gracias al uso de algoritmos de *Deep Learning* para la identificación de ganado bovino a partir de los puntos que presenta el hocico de los animales [8]. Los autores utilizaron las mismas 5000 imágenes de los hocicos evaluados previamente; sin embargo, para realizar el aprendizaje de las características discriminantes sobre las texturas, utilizaron los algoritmos *Convolutional Neural Networks* (CNN) [20], *Deep Belief Networks* (DBN) [21] y *Stacked Denoising Autoencoders* [22], obteniendo una precisión de identificación del 75.98 %, 88.46 % y 95.99 % respectivamente.

La zoometría estudia las formas de los animales mediante mediciones corporales concretas que permiten cuantificar la conformación corporal. De esta forma, es posible conocer a través de las diferencias, la existencia de biotipos dentro de una raza o la inclinación hacia determinada producción zootécnica [23]. La zoometría es considerada como un elemento de

trabajo importante a la hora de definir una población (sea para un morfotipo, paratipo o prototipo), así como marcar tendencias productivas o deficiencias zootécnicas. Por ejemplo, en [7] se desarrolló un software para aplicar zoometría en bovinos. Dicha investigación proporciona una herramienta tecnológica para acortar notablemente la subjetividad de la decisión, pues por medio de dicha aplicación se captura, compara y estandariza un lote de animales, determinando cuál de éstos animales cuenta con potencial de mejora genética en comparación con la generación actual. Se analizaron 30 hembras adultas, de la raza zoogenética nativa Blanco Orejinegro. Se tomaron las fotografías en 2D desde diferentes ángulos de medición manteniendo la misma distancia con respecto al animal. Luego se procedió a capturar las diferentes medidas. Las variables están representadas en 16 medidas de alzada y longitud, 8 medidas de anchura y longitud, y 6 medidas cefálicas en las cuales se analizaron índices etnológicos y funcionales.

Basado en la técnica de zoometría, en el presente trabajo se logró construir una aplicación que permite una selección mucho más objetiva que la selección por observación directa; ya que se emplea registros individuales de datos con diferentes medidas por animal. Para identificar aquel animal que se aproxima a la medida ideal, el usuario carga la imagen, con las respectivas medidas que se esperaría en un animal ideal. Estas medidas son confrontadas a nivel de grupo para poder extraer los animales que se aproximen a dichas medidas.

El estudio de la literatura nos muestra varios intentos de construir sistemas para la clasificación de ganado usando técnicas de procesamiento de imágenes, sin embargo, pocos han utilizado las características corporales del animal.

III. DESCRIPCIÓN DEL CONJUNTO DE DATOS

En la presente sección se describe el conjunto de datos que fueron colectados y pre-procesados para el entrenamiento y validación de nuestro modelo.

El conjunto de 350 imágenes de vacas multíparas (mayor a dos partos) adquiridas en la granja de la Unidad Experimental de Zootecnia de la Universidad Nacional Agraria La Molina en Lima, Perú contienen 139 imágenes del tren posterior y 211 imágenes del perfil del animal. Dichas imágenes fueron adquiridas con cámaras de los teléfonos móviles de los participantes del proyecto. Estos teléfonos son de gama media, con cámaras de 12 megapíxeles de resolución en promedio. Las únicas restricciones en el protocolo de la captura de datos fueron: 1) las imágenes deben ser tomadas de día; 2) que en la imagen solo aparezca un animal; y, 3) que la fotografía capture toda la vaca (no solo una parte de ella). No es imprescindible que las imágenes del tren posterior y del perfil pertenezcan al mismo animal. Cabe resaltar que las 350 imágenes fueron divididas en dos grupos: 200 imágenes para la etapa de entrenamiento; y, 2) 150 imágenes para la etapa de validación. De esta manera, las imágenes de entrenamiento servirán para construir un modelo de clasificación, y las imágenes de validación nos permitirán medir cuán eficiente es este modelo.

En lo que respecta a la evaluación fenotípica, esta la realizó un especialista del establo, basándose en el modelo de evaluación morfológica internacional del vacuno de leche de la World

Holstein-Friesian Federation (WHFF), para dos características descriptivas lineales [24]: profundidad de la ubre y la anchura de la grupa; las cuales se describen a continuación.

Por un lado, para determinar la profundidad de la ubre se evaluó la medida entre los corvejones y la parte más baja del piso de la ubre como se muestra en la Fig. 1. Se calificó con 2 si el experto consideraba que la ubre se encontraba por debajo del corvejón, con 5 si consideraba que se encontraba al nivel del corvejón y con 10 si consideraba que se encontraba por encima del corvejón.

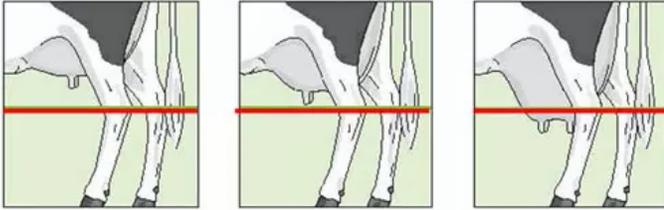


Fig. 1. Evaluación lineal de la profundidad de la ubre [24]

Por otro lado, para medir la anchura de la grupa se tomó la distancia entre los isquiones. Se calificó con 3 si el experto consideraba que era muy estrecha, con 6 si consideraba que era intermedia y con 9 si consideraba que era ancha, tal como se visualiza en la Fig. 2.

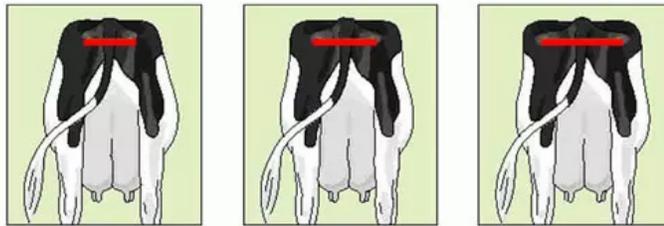


Fig. 2. Evaluación lineal de la anchura de la grupa [24]

Para calcular los índices descritos anteriormente, se utilizaron las imágenes de la Fig. 3, que sirvieron como plantillas. Una de ellas refleja el perfil para la evaluación de la profundidad de la ubre, y la otra que refleja el tren posterior para la evaluación del ancho de la grupa.

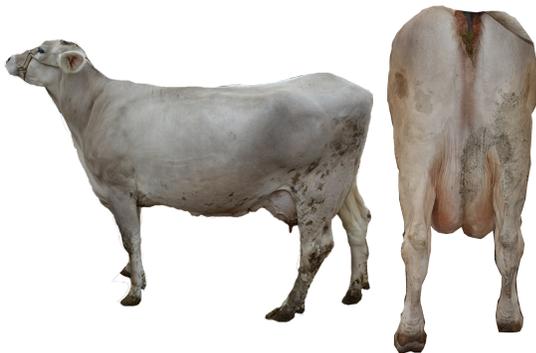


Fig. 3. Parte perfil y tren posterior del vacuno luego de la etapa de segmentación manual usando la herramienta GIMP.

IV. METODOLOGÍA Y EXPERIMENTOS

En esta sección se detalla el proceso propuesto, así como los experimentos realizados para validar el modelo desarrollado. Así, proponemos la creación de un modelo para la detección y seguimiento de vacas y el desarrollo de un motor de soporte para la toma de decisiones, que permita la identificación y clasificación automática del ganado de la raza Brown Swiss. Este proceso consta de dos partes, la primera de detección y seguimiento, y la segunda de soporte para la evaluación fenotípica (*c.f.*, Fig. 4). Más en detalle, la primera fase consiste en detectar el ganado vacuno. Una vez detectado el animal, procedemos a visualizar la plantilla sobre el área de interés detectada (*c.f.*, Fig. 4 (A)). Esta plantilla ayudará al usuario a realizar la evaluación sobre la característica fenotípica. En la segunda fase ilustrada en la Fig. 4 (B), el motor de inferencia nos permitirá determinar, según el puntaje acumulado por la elección de la plantilla, si es o no el animal de la raza Brown Swiss. Finalmente, podremos visualizar los resultados de la evaluación.

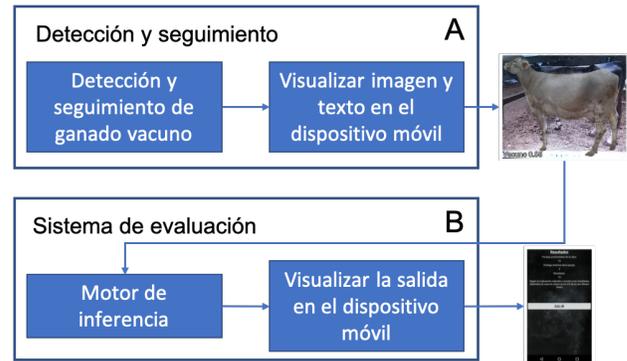


Fig. 4. Modelo de dos etapas: A) detección y seguimiento; y, B) sistema de evaluación.

A continuación detallamos como se implementaron la detección de objetos y el motor de inferencia.

En lo que concierne a la detección de objetos, se utilizó la API de detección de objetos de TensorFlow [25], el cual es un framework de código abierto que hace fácil la construcción, entrenamiento y despliegue de los modelos generados para detectar objetos [26]. Asimismo, se eligió el modelo MobileNet [27], por su velocidad de intercambio, ya que es más ligero y tiene buenos resultados si se quiere desarrollar aplicaciones para dispositivos móviles. Además, se utilizó como herramienta de desarrollo Android Studio, para el desarrollo de la aplicación móvil. Con respecto a la segmentación manual de imágenes, se utilizó la herramienta GIMP, para eliminar el fondo de cada imagen que servirá como plantilla al momento de evaluar al ganado vacuno. Por otro lado, se utilizó la herramienta RectLabel para delimitar la zona de interés en la imagen y definir las clases de objetos a detectar. Finalmente, se utilizó el lenguaje de programación Python con TensorFlow, para generar el modelo utilizando el API de detección de objetos.

Para generar el modelo, se realizó la anotación de las imágenes, la delimitación de la zona de interés y la definición de las clases de objeto a detectar para cada imagen. Para ello

TABLA I
CARACTERÍSTICAS DE LAS IMÁGENES PREVIAS AL
PROCESO DE ENTRENAMIENTO.

Filename	Width	Height	Class	XMin	YMin	XMax	YMax
DSC_0204.JP	6000	4000	Vacuno	17	438	5293	3999
DSC_0205.JP	6000	4000	Vacuno	250	269	5517	3816
DSC_0206.JP	6000	4000	Vacuno	270	269	5530	3769
DSC_0207.JP	6000	4000	Vacuno	1390	1	4303	3889
DSC_0208.JP	6000	4000	Vacuno	704	13	3986	5913
DSC_0209.JP	6000	4000	Vacuno	1186	586	3495	5822
DSC_0210.JP	6000	4000	Vacuno	613	22	3131	5568
DSC_0211.JP	6000	4000	Vacuno	3	16	5993	3999
DSC_0212.JP	6000	4000	Vacuno	10	9	5988	3999
DSC_0213.JP	6000	4000	Vacuno	543	396	5457	3829
DSC_0214.JP	6000	4000	Vacuno	577	269	5543	3823
DSC_0215.JP	6000	4000	Vacuno	3	3	5537	3996
DSC_0216.JP	6000	4000	Vacuno	550	116	5217	3923
DSC_0217.JP	6000	4000	Vacuno	443	69	5130	3849
DSC_0218.JP	6000	4000	Vacuno	740	686	4000	5442

se utilizó la herramienta RectLabel, delimitando el área del objeto a detectar y definiendo la clase como “Vacuno”, tal como se observa en la Fig. 5.

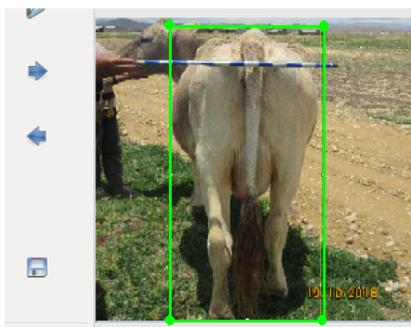


Fig. 5. Preparación de las imágenes previo al entrenamiento.

La Tabla I. muestra la información generada en el proceso anterior, tal como, el ancho, la altura, la clase y las posiciones máximas y mínimas de la zona delimitada. Finalmente, se convirtió el archivo que contiene las anotaciones de las imágenes a un formato *.record*, a fin de que TensorFlow lo pueda usarlo como entrada en el proceso de entrenamiento del modelo.

En lo que concierne el entrenamiento del modelo, muchos algoritmos en la literatura han sido utilizados para la tarea de clasificación de objetos a partir de imágenes (ver Tabla II). Así, las aplicaciones móviles que requieren aplicar clasificación, detección o segmentación de imágenes en tiempo real utilizan servidores de calculo remotos que procesan la respuesta para enviar los resultados a la aplicación en al dispositivo móvil como un teléfono celular (*i.e.*, arquitectura cliente/servidor). Justamente, la primera versión de nuestra aplicación utilizó esta tecnología, donde el cliente capturaba una imagen y la enviaba - vía una conexión de red - al servidor para que fuera procesada y el resultado era reenviado al cliente. Por el lado del servidor, probamos los algoritmos de KNN [12], y Redes Neuronales [16], sin embargo, la aplicación no cumplía con el objetivo de portabilidad ya que era dependiente de una conexión a Internet. Después, optamos por utilizar modelos pre-entrenados, los cuales son rápidos, ligeros para ser usados en equipos móviles y no necesitan de una conexión una vez

el modelo ha sido construido.

TABLA II
ALGORITMOS DE ENTRENAMIENTO PROBADOS.

Modelo	Portable	Referencia
KNN	No	[12]
ANN	No	[16]
Transfer learning	Si	[28]

Precisamente, en la fase de construcción de un modelo, se utilizó el enfoque por transferencia de aprendizaje o *transfer learning* [28], cuya implementación se encuentra en la herramienta Mobilenet de la plataforma Tensorflow. Durante el entrenamiento, se ajustó el modelo previamente entrenado, proporcionando nuevas imágenes al análisis, las cuales fueron categorizadas finalmente como clase “Vacuno”. Posteriormente se realizaron algunos ajustes a la red (Hiperparámetros: *batch_size* = 24, *initial_learning_rate* = 0.004, *num_steps* = 600), para categorizar sólo esta clase. La Tabla III, muestra los resultados obtenidos por Mobilenet, en la cual, podemos observar el tamaño del modelo (17MB), la exactitud mínima de 0.665 y la máxima de 0.871. Cabe recalcar que la exactitud se refiere al rendimiento del modelo en el conjunto de imágenes de validación obtenidos del repositorio de ImageNet [29].

TABLA III
MODELO UTILIZADO PRE-ENTRENADO EN EL CONJUNTO
IMAGENET.

Modelo	Tamaño	Exactitud mínima	Exactitud máxima
MobileNet	17MB	0.665	0.871

En nuestro estudio la evaluación tomando el modelo pre-entrenado se hizo directamente en el teléfono sin comprometer sus recursos y respuestas. Para ello, en la etapa de detección del ganado vacuno se utilizó el API de detección de TensorFlow. Es necesario recordar que hemos elegido este *framework* por su flexibilidad, control y funcionalidad, tal como se menciona en [30]. El API de detección de objetos de TensorFlow está compuesto por las siguientes siete etapas que se ilustran en la Fig. 6. 1) La descarga del conjunto de datos, que permitió afinar el modelo pre-entrenado utilizando imágenes de ganado vacuno; 2) la anotación de imágenes, que consistió en delimitar la zona de interés en la imagen y en definir las clases de objeto a detectar; 3) la preparación del mapa de etiquetas, que consistió en asociar una etiqueta por cada clase; 4) la generación de archivos *.record*, formato que TensorFlow puede interpretar y que contienen la información del etiquetado por clase: ancho, alto, profundidad, puntos mínimos y máximos de la zona de interés; 5) la configuración de los hiper-parámetros: *batch_size* = 24, *initial_learning_rate* = 0.004, *num_steps* = 600; 6) el entrenamiento del modelo; y, 7) la inferencia, que es la etapa en el que los cuadros delimitadores se dibujaron en *frames* detectados para ser combinados posteriormente [31].

Una vez probado el modelo, integramos la herramienta para los usuarios finales. El motor evaluación fenotípica está asociado a una base de datos con valoración fenotípica lineal.



Fig. 6. Fases de implementación del API de detección de objetos de TensorFlow [31].

Una vez detectado el ganado vacuno, las plantillas se superponen sobre el área de interés encontrado; permitiendo de esta manera al usuario seleccionar la plantilla que más se adecúe al animal. Por ejemplo, la Fig. 7, muestra la superposición de la plantilla sobre la vaca detectada. En esta plantilla se analiza la vista lateral de la vaca para medir la profundidad de la ubre, mediante la aparición de tres plantillas generadas por el sistema y la selección de la más adecuada por parte del ganadero. En esta forma de evaluación, el ganadero puede realizar la evaluación fenotípica *in-situ* y la pericia del evaluador se refleja en las plantillas previamente construidas. Finalmente, en la Fig. 7 se muestra la tasa de acierto del proceso de identificación del ganado vacuno. La selección de la plantilla que más se adecúe al animal le da un puntaje en función de las medidas propuestas [24]. Este proceso permitiría determinar, según el puntaje acumulado, si es o no el animal de la raza Brown Swiss y asignar una puntuación en función de la silueta que el experto elija.



Fig. 7. Superposición de una de las plantillas sobre la vaca detectada.

V. RESULTADOS

En la presente sección se muestran los resultados obtenidos con el modelo descrito en la sección previa.

La Fig. 8, muestra los resultados de la evaluación fenotípica del ganado vacuno de la raza Brown Swiss, mediante la detección, seguimiento y soporte de toma de decisiones. Los resultados de la evaluación se muestran acorde al puntaje asignado a cada plantilla generada.

Con respecto al desempeño de nuestro modelo, la Fig. 9, muestra las gráficas de precisión y pérdida del entrenamiento. Se tomó en cuenta la medida *Mean Average Precision* (mAP), la cual es la precisión obtenida versus el número de pasos en el entrenamiento. Además, utilizamos la Intersección sobre la Unión (IoU) también conocido como índice de Jaccard,



(a)

(b)



(c)

Fig. 8. Aplicación móvil: a) evaluación sobre la profundidad de la ubre; b) evaluación sobre la anchura de la grupa; y c) resultado de la evaluación.

que mide el grado de similitud existente entre dos conjuntos. En este trabajo utilizaremos el IoU para medir la precisión del clasificador. Esta medida nos indica qué tan sensible es la red para detectar los objetos de interés (ganado). Cuanto más alta es la puntuación de mAP, más precisa es la red. En general, en la Fig. 9 se observa que la precisión aumenta mientras que la pérdida disminuye por cada iteración en la fase de entrenamiento. Las líneas azules fuertes representan el entrenamiento realizado, mientras que las líneas azules claras representan los valores máximos a los que se pueden llegar en esta fase de entrenamiento del modelo basado en la verdad de terreno. De forma mas detallada, la Fig. 9 (A) muestra la precisión versus el número de pasos del entrenamiento. La primera gráfica muestra la precisión promedio, la segunda gráfica la puntuación IoU al 50 % y la tercera gráfica la puntuación IoU al 75 % del total de imágenes evaluadas. Por otro lado, la Fig. 9 (B) muestra la exhaustividad versus el número de pasos del entrenamiento, con 1, 10 y 100 detecciones respectivamente. Finalmente, la Fig. 9 (C) muestra la pérdida generada por TensorFlow, la cual asciende a 7.282.

Como resumen de lo anteriormente dicho, la Tabla IV muestra la precisión promedio, la puntuación IoU al 50 % y la puntuación IoU al 75 % para nuestro modelo.

Como los resultados preliminares obtenidos demostraban una baja precisión promedio del modelo desarrollado (*c.f.*

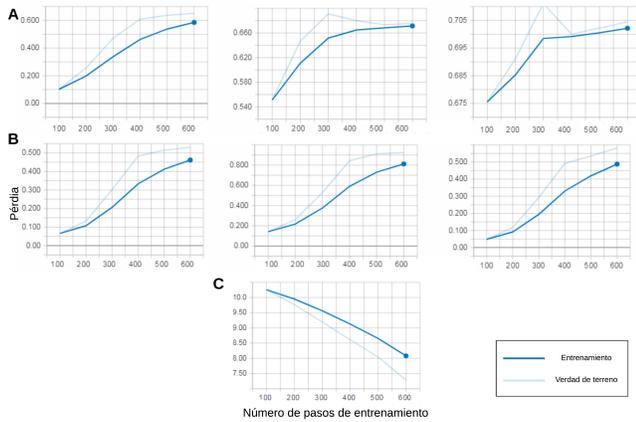


Fig. 9. Gráficas de rendimiento generados por TensorFlow.

TABLA IV
PRECISIÓN PROMEDIO (AP) DEL MODELO.

	AP	AP50IoU	AP75IoU
ModeloSSD	0.5294	0.9241	0.5822

Tabla III), principalmente por el uso de imágenes en el que figuraban animales Brown Swiss de manera aislada, se realizó un pre-entrenamiento complementario adicional con imágenes en los que se encontraban personas, animales de otras razas, entre otros objetos. Para ello utilizamos la herramienta *SSD: Single Shot MultiBox Detector* [32].

Así, el nuevo modelo generado, obtuvo una recuperación promedio con una detección de 0.6511, con diez detecciones de 0.6756, y con cien detecciones de 0.7044 (*c.f.*, Tabla V).

TABLA V
EXHAUSTIVIDAD PROMEDIO (AR) DEL NUEVO MODELO.

	AR1	AR10	AR100
ModeloSSD	0.6511	0.6756	0.7044

Así vemos que el modelo planteado que se baso en MobileNet es capaz de detectar objetos “vacunos” entre 65 % a 70 % de las veces.

VI. CONCLUSIONES Y TRABAJOS FUTUROS

El presente estudio ha logrado construir una aplicación móvil para la automatización del proceso general de evaluación de las características fenotípicas del ganado vacuno de la raza Brown Swiss, superponiendo plantillas a través del desarrollo de un modelo de detección de vacas.

Se ha podido comprobar que realizar la detección de objetos en tiempo real se vuelve más ligero y eficiente si utilizamos modelos pre-entrenados, como MobileNet, el cual es un modelo eficiente para móviles y aplicaciones con pocos recursos [27]. Con respecto a la elección del modelo de aprendizaje, podemos elegir modelos pre-entrenados según la precisión y la velocidad de intercambio. La API de detección de objetos de TensorFlow nos permite la construcción, entrenamiento y despliegue de estos modelos. En este sentido, se eligió

el modelo MobileNet, por su velocidad de intercambio en sacrificio de la precisión. Sin embargo, para mayor precisión comprometiendo la velocidad, podemos utilizar redes como Xception [33], Inception [34], VGG16 [35], VGG19 [35], ResNet [35], entre otras.

Con respecto a la evaluación de la eficacia de los algoritmos de aprendizaje, el mejor resultado obtenido en puntuación IoU del nuevo modelo generado es de 0.9241 al evaluar el 50 % del total de imágenes. El mejor resultado de exhaustividad es 0.7044 con 100 detecciones, lo que quiere decir que se ha recuperado el 70 % del total de imágenes encontradas en la base de datos.

Se sugiere que los trabajos futuros se centren básicamente en la experimentación con otros algoritmos de detección de características y comparar la eficacia y rapidez en el proceso de detección. También se propone la evaluación fenotípica ya no bajo plantillas sino realizando la toma de medidas en tiempo real.

REFERENCIAS

- [1] Portal del Ministerio de Agricultura y Riego, “Vacunos de leche,” <http://minagri.gob.pe/portal/40-sector-agrario/situacion-de-las-actividades-de-crianza-y-produccion/303-vacunos-de-leche?start=16,2019>, online; accessed Feb-2019.
- [2] F. Madalena, “Considerations on the management of animal genetic resources in latin america,” in *Proceedings of the EAAP/SLU/FAO/ICAR Workshop. Uppsala, Sweden. June, vol. 2, 2005*.
- [3] J. Corrales, M. Cerón-Muñoz, A. Jhon Cañas, C. Herrera, and C. Samir Calvo, “Relationship between type traits and milk production in holstein cows from antioquia, colombia,” *Revista MVZ Córdoba*, vol. 16, no. 2, pp. 2507–2513, 2011.
- [4] Holstein Association USA, “Características descriptivas lineales,” www.holsteinausa.com, online; accessed Feb-2019.
- [5] S. Kumar, S. K. Singh, R. Singh, and A. K. Singh, *Animal Biometrics: Techniques and Applications*. Springer, 2018.
- [6] G. Ronghua, G. JingQiu, and L. Jubao, “Cow behavioral recognition using dynamic analysis,” in *2017 International Conference on Smart Grid and Electrical Automation (ICSGEA)*. IEEE, 2017, pp. 335–338.
- [7] J. P. Rincón, C. A. U. Castro, and A. R. Gómez, “Desarrollo de un software para aplicar zoometria en bovinos,” *Revista Colombiana De Tecnologías De Avanzada (RCTA)*, vol. 1, no. 25, 2017.
- [8] S. Kumar, A. Pandey, K. S. R. Satwik, S. Kumar, S. K. Singh, A. K. Singh, and A. Mohan, “Deep learning framework for recognition of cattle using muzzle point image pattern,” *Measurement*, vol. 116, pp. 1–17, 2018.
- [9] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, vol. 1. IEEE, 2005, pp. 886–893.
- [10] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *European conference on computer vision*. Springer, 2006, pp. 404–417.
- [11] C. Silva, T. Bouwmans, and C. Frélicot, “An extended center-symmetric local binary pattern for background modeling and subtraction in videos,” 2015.
- [12] J. M. Keller, M. R. Gray, and J. A. Givens, “A fuzzy k-nearest neighbor algorithm,” *IEEE transactions on systems, man, and cybernetics*, no. 4, pp. 580–585, 1985.
- [13] J. Park and I. W. Sandberg, “Universal approximation using radial-basis-function networks,” *Neural computation*, vol. 3, no. 2, pp. 246–257, 1991.
- [14] J. R. Quinlan, “Induction of decision trees,” *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.
- [15] D. Reynolds, *Gaussian Mixture Models. Encyclopedia of Biometrics* 827–832, 2015.
- [16] A. Luna, M. Nunez-del Prado, A. Talavera, and E. S. Holguín, “Power demand forecasting through social network activity and artificial neural networks,” in *2016 IEEE ANDESCON*. IEEE, 2016, pp. 1–4.
- [17] D. F. Specht, “Probabilistic neural networks,” *Neural networks*, vol. 3, no. 1, pp. 109–118, 1990.

- [18] S. Haykin and N. Network, "A comprehensive foundation," *Neural networks*, vol. 2, no. 2004, p. 41, 2004.
- [19] H. Zhang, "Exploring conditions for the optimality of naive bayes," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, no. 02, pp. 183–198, 2005.
- [20] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Jun. 2014, pp. 655–665.
- [21] G. E. Hinton, "Deep belief networks," *Scholarpedia*, vol. 4, no. 5, p. 5947, 2009.
- [22] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of machine learning research*, vol. 11, no. 12, 2010.
- [23] T. Vera and A. Ricarte, "Procedimientos para la caracterización zoológica y fanerótica de las razas o biotipos raciales existentes en las majadas caprinas de la rioja," 10 2010.
- [24] L. Aguirre, "Characteristics to select in dairy cattle for livestock in southern region of ecuador," *Revista Centro de Biotecnología*, vol. 2, pp. 60–67, 01 2013.
- [25] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 2016, pp. 265–283.
- [26] J. Huang and V. Rathod, "Supercharge your computer vision models with the tensorflow object detection api," *Google AI Blog*, vol. 15, 2017.
- [27] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv*, 2017.
- [28] A. Oliva Rodríguez, "Desarrollo de una aplicación de reconocimiento en imágenes utilizando deep learning con opencv," Ph.D. dissertation, 2018.
- [29] J. Eckroth, *A Blueprint for Detecting Your Logo in Social Media in AI Blueprints: How to build and deploy AI business projects*. Packt Publishing Ltd, 2018.
- [30] A. Sergeev and M. D. Balso, "Horovod: fast and easy distributed deep learning in tensorflow," *arXiv*, 2018.
- [31] T. V. Janahiraman and M. S. M. Subuhan, "Traffic light detection using tensorflow object detection framework," in *2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*. IEEE, 2019, pp. 108–113.
- [32] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.
- [33] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.
- [34] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, 2017.
- [35] L. Wen, X. Li, and L. Gao, "A transfer convolutional neural network for fault diagnosis based on resnet-50," *Neural Computing and Applications*, pp. 1–14, 2019.



Julianna Milagros Apumayta Lopez Estudiante de Maestría en Ingeniería Informática con Mención en Ciencias de la Computación de la Pontificia Universidad Católica del Perú. Bachiller en Ingeniería de Sistemas de la Universidad Nacional de San Agustín de Arequipa. Joven investigadora con experiencia en proyectos de Innovación y desarrollo de aplicaciones para el sector financiero y de banca. Sus áreas de interés en investigación son: Procesamiento de imágenes, computación móvil y ciencia de datos.



Eduardo Leuman Fuentes Navarro Ingeniero Zootecnista egresado de la UNALM, Mg. Sc. en Nutrición y Desarrollo Rural de la Universidad de Gent (Bélgica) y Doctor de la Universidad de Cork (Irlanda) y Montpellier Supagro (Francia) en el marco del programa europeo Erasmus Mundus "Agricultural Transformation by Innovation". Se desempeña como docente visitante en los programas doctorales de Ciencia Animal y de Economía de los Recursos Naturales y Desarrollo Sustentable de la UNALM. Especialista en innovación y transferencia de tecnología, desarrollo sustentable, modelización de sistemas de producción, mitigación y adaptación frente al cambio climático, evaluación de pastizales, mejoramiento genético de ganado, elaboración de productos lácteos y evaluación de calidad de leche.



Miguel Nuñez del Prado Es doctor en informática por la Universidad de Toulouse. Obtuvo este título por su trabajo sobre ataques de inferencia en datos geolocalizados y su impacto en la privacidad de los usuarios en el LAAS-CNRS Francia. Es ingeniero en Computación, Redes y Telecomunicaciones. Tiene dos maestrías, una en Informática y Telecomunicaciones y otra en Gestión estratégica de la Innovación. Trabajó como científico de datos en el Grupo INTERSEC (París, Francia).



Hugo Alatrística Salas Doctor en Ciencias de la Computación de la Universidad de Montpellier en Francia. El tema de investigación con el cual obtuvo su título está relacionado con la Minería de Datos espacio-temporal con aplicación en el medio ambiente y la salud pública. Además, tiene una maestría en Calculabilidad, Algorítmica, Seguridad y Administración de Redes de la misma Universidad. Actualmente, es profesor investigador en la Universidad del Pacífico y vice decano del programa de ingeniería de la Información.