

## Article

# Computer Vision-Based Approach for Automatic Detection of Dairy Cow Breed

Himanshu Gupta <sup>1</sup>, Parul Jindal <sup>1</sup>, Om Prakash Verma <sup>1</sup>, Raj Kumar Arya <sup>2</sup>, Abdelhamied A. Ateya <sup>3</sup>, Naglaa. F. Soliman <sup>4,\*</sup> and Vijay Mohan <sup>5,\*</sup>

- <sup>1</sup> Department of Instrumentation and Control Engineering, Dr. B. R. Ambedkar National Institute of Technology Jalandhar, Jalandhar 144027, India
- <sup>2</sup> Department of Chemical Engineering, Dr. B. R. Ambedkar National Institute of Technology Jalandhar, Jalandhar 144027, India
- <sup>3</sup> Department of Electronics and Communications Engineering, Zagazig University, Zagazig 44519, Egypt
- <sup>4</sup> Department of Information Technology, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
- <sup>5</sup> Department of Mechatronics, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India
- \* Correspondence: nfsoliman@pnu.edu.sa (N.F.S.); vijay.mohan@manipal.edu (V.M.)

**Abstract:** *Purpose:* Identification of individual cow breeds may offer various farming opportunities for disease detection, disease prevention and treatment, fertility and feeding, and welfare monitoring. However, due to the large population of cows with hundreds of breeds and almost identical visible appearance, their exact identification and detection become a tedious task. Therefore, the automatic detection of cow breeds would benefit the dairy industry. This study presents a computer-vision-based approach for identifying the breed of individual cattle. *Methods:* In this study, eight breeds of cows are considered to verify the classification process: Afrikaner, Brown Swiss, Gyr, Holstein Friesian, Limousin, Marchigiana, White Park, and Simmental cattle. A custom dataset is developed using web-mining techniques, comprising 1835 images grouped into 238, 223, 220, 212, 253, 185, 257, and 247 images for individual breeds. YOLOv4, a deep learning approach, is employed for breed classification and localization. The performance of the YOLOv4 algorithm is evaluated by training the model on different sets of training parameters. *Results:* Comprehensive analysis of the experimental results reveal that the proposed approach achieves an accuracy of 81.07%, with maximum *kappa* of 0.78 obtained at an image size of 608 × 608 and an intersection over union (*IoU*) threshold of 0.75 on the test dataset. *Conclusions:* The model performed better with YOLOv4 relative to other compared models. This places the proposed model among the top-ranked cow breed detection models. For future recommendations, it would be beneficial to incorporate simple tracking techniques between video frames to check the efficiency of this work.

**Keywords:** automatic livestock farming; cow breed classification; deep learning; object detection; YOLOv4



**Citation:** Gupta, H.; Jindal, P.; Verma, O.P.; Arya, R.K.; Ateya, A.A.; Soliman, N.F.; Mohan, V. Computer Vision-Based Approach for Automatic Detection of Dairy Cow Breed. *Electronics* **2022**, *11*, 3791. <https://doi.org/10.3390/electronics11223791>

Academic Editor: Silvia Liberata Ullo

Received: 27 October 2022

Accepted: 16 November 2022

Published: 18 November 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Animal husbandry is one of the most lucrative and demanding businesses worldwide and contributes significantly to the nation's gross domestic product (GDP). As per the report published by the World Bank (2022), agriculture (and its allied sectors) accounts for almost 4.01% of the world's GDP, which in developing countries significantly increases up to 25% [1]. Figure 1 represents India's GDP distribution, showing that agriculture contributes nearly 19% of the GDP [2]. In particular, dairy farming contributes majorly (about 5.30%), with milk as the significant livestock product [3]. As per the report published in 2020 by Indian National Accounts Statistics (NAS), the livestock sector contributes 4.19% of the total gross value added (GVA) and 28.63% of the total agriculture and allied sector GVA [2]. These businesses are majorly governed by small, peripheral farmers and landless

workers. Dairy farming is a secondary source of income for thousands of rural families and provides livelihood to two-thirds of the rural community. This sector is undergoing rapid growth due to urbanization, population growth, and, most importantly, the rise in income in developing countries. Further, the Indian policymakers also suggested self-sustainable models empowered with the more significant employment of technologies and market linkage to double farmers' income by 2022 [4].

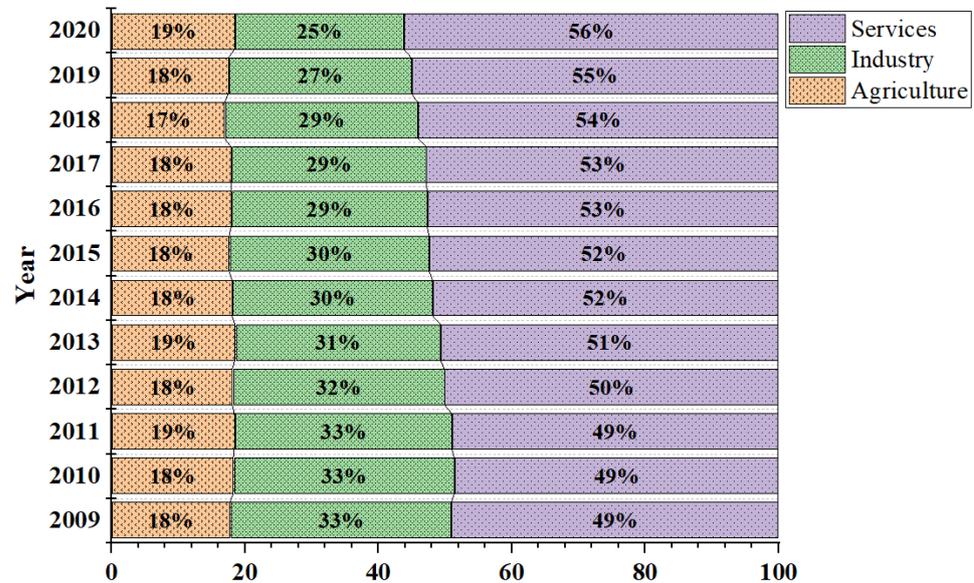


Figure 1. Distribution of GDP across economic sectors in India.

### 1.1. Motivation

Proficient animal husbandry would allow stockmen (and eventually the associated companies or national agencies) to earn more profit. Their commercial value highly influences domestic cattle profit (particularly cows) and the cost to raise them. The mercantile value of cows mainly depends on their fertility rate, milk production, and the chemical composition of milk. Moreover, different breeds yield different milk varieties, since breed affects milk composition, fatty acid composition, and coagulation properties [5,6]. Due to the different stages of lactation in breeds, milk production is similar for one breed of cow but different from one breed to another. For instance, Gyr cows produce 900–1600 kg of milk, whereas Holstein Friesian produce 7200–9000 kg per lactation [7]. This variation in milk yield between individual cows can turn into significant losses for businesses that encompass thousands of cows. Therefore, identifying a cow's breed would benefit the dairy industry.

Due to the increase in population and decrease in farms, dairy livestock needs better monitoring for breed associations. Therefore, identifying the breed of individual cattle is a key to dairy farming. Breed identification of individual cows may offer information to the stockmen and assist in making important decisions about that animal, such as the opportunity for cross-breeding to enhance the production rate. Additionally, recognizing breeds plays a vital role in automatic behavior analysis, health monitoring, and the detection of lameness and helps estimate their fertility rate. Further, the individual identification and tracking of cow breeds may offer various farming opportunities for disease detection (e.g., early detection of disease outbreaks and transmission), disease prevention and treatment, fertility and feeding, and welfare monitoring. Identification of cow breeds would balance the trade-off between cost and management, thereby improving the productivity and profitability of dairy farms.

## 1.2. Related Work

Generally, breed identification methods have been classified into tag-based and visual-feature-based approaches. The tag-based methods use permanent markings (such as tattooing and ear notching), temporary markings (such as ear tagging), and electronic devices (such as radio frequency identification, RFID) [8]. The permanent markings can only be applied to identify individuals in smaller groups, whereas the temporary markings have been found susceptible to fraudulent manipulation and easy duplication [9]. Furthermore, these approaches require specific sensing devices on the body, which may require invasive techniques. These bottlenecks led to the development of RFID-based electronic identification devices. However, implementing RFID chips and scanners at various checkpoints is a challenging task and requires skilled persons, particularly while monitoring groups of animals. Moreover, these methods are prone to duplications or false identification when monitoring numerous livestock animals in harsh outdoor environments. In addition, these devices are expensive and can be easily damaged.

The reported literature reveals various visual-biometric-based cattle identification methodologies that utilize the unique external biometric characteristics of breeds (such as coat pattern, muzzle pattern, and body contour) to effectively address the limitations of tag-based techniques [10]. However, the exact identification and extraction of these features are challenging, even for an expert, which limits the wide acceptance of earlier approaches. In contrast, deep learning (DL) techniques have established their sovereignty for complex object detection and recognition tasks [11,12]. Motivated by this, DL approaches have been successfully employed to extract the hidden features to classify and localize species such as sheep, dogs, and birds [13–15]. Similar trends have been witnessed in preserving cow breeds for the state's cultural and genetic heritage [16].

Among various visual features, muzzle patterns have been widely used for cattle identification because of the distinct grooves and beaded prints [17]. For example, DL methodologies have been employed to extract distinguishable features from muzzle images [18,19]. In another work, an auto-encoder and a deep belief network have been used to find hidden features of cow nosea [20]. However, this approach ignored information about other essential body parts, such as head and legs, resulting in reduced accuracy. Further, retinal features have been used to identify cattle, but the difficulties in capturing livestock retinal images limit its practical applicability [21]. Additionally, the hidden patterns of body coat and face have been exploited to identify individual cattle [22]. Meanwhile, incorporating convolutional neural networks (CNNs) ensures the automatic extraction of rich features, resulting in the improved identification of cattle breeds [23]. Later, beef cattle were detected in image sequences by fusing CNN and long short-term memory (LSTM) algorithms [24]. Another pioneering work automatically detects Holstein Friesian cattle by extracting coat pattern features [25,26]. Further, it utilizes DL techniques (CNN and a recurrent convolutional network, RCN) on the ground and aerial view images for cattle breed identification. Similarly, computer vision techniques such as you only look once (YOLO) and region-based CNN (RCNN) have been employed to detect cattle breeds using their morphological features [27,28]. However, these models were limited to detecting only one breed and lacked emphasis on breed diversity. Table 1 summarizes the DL-based previously reported cow breed detection models.

Based on the literature mentioned above, we identify the following research gaps:

1. Although the outcomes of the invasive techniques are promising, the experimental designs have certain flaws that make it difficult to evaluate the real significance of the reported results in harsh environments.
2. Cattle have previously been recognized based on the characteristics of specific body parts. However, other key body components such as the head and legs were left out, which may result in the loss of crucial information.
3. In most of the literature, the emphasis is on identifying a single breed. However, these approaches cannot identify and classify various breeds.

**Table 1.** Deep-learning-based cow breed detection models.

Authors	Number of Breeds	Dataset Size	Model	Performance (%)
Tassinari et al. [27]	1	11,754 images	YOLOv3	66.00 (P)
Andrew et al. [28]	1	4736 images	YOLOv3	92.40 (A)
Andrew et al. [26]	1	32 videos	YOLOv2	91.90 (A)
Andrew et al. [25]	1	34 videos	Faster RCNN	98.13 (A)
Bello et al. [20]	1	4000 images	DBN + auto-encoder	94.55 (A)

P: precision, A: accuracy, DBN: deep belief network.

### 1.3. Contributions

To our knowledge, there is no model for the automatic detection of cow breed in animal biometrics literature that bridges the research gaps mentioned above. Therefore, there is a need to cater to improved and advanced methods for detecting cattle breeds based on overall body features. Thus, the present work presents the first proof-of-concept system for automatic cow breed detection. To summarize, the main contributions of this study are:

1. Development of a multi-breed cow detection framework based on YOLOv4 to identify and classify diverse cow breeds with high accuracy;
2. Development of a custom cow dataset containing multiple breeds using web-mining techniques;
3. Comparative performance analysis of simulated results to endorse the most prominent training parameters.

### 1.4. Structure of the Paper

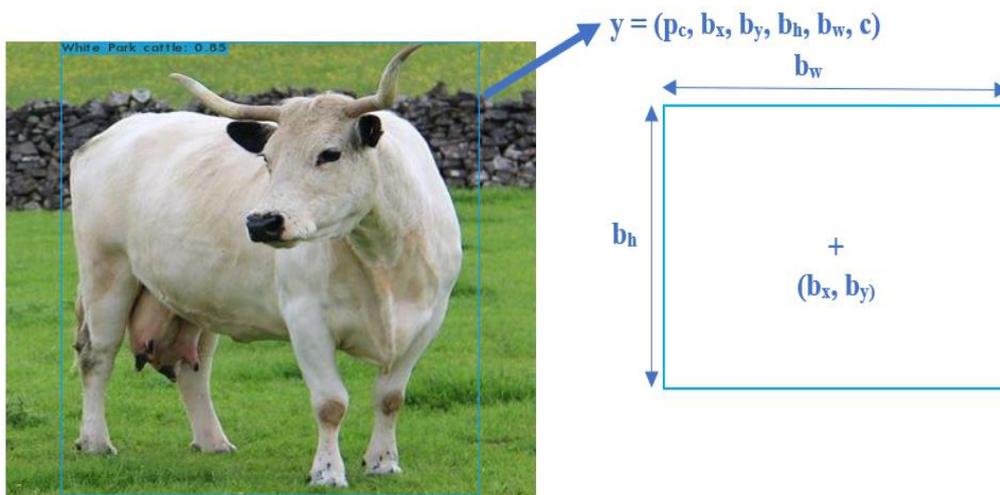
The remaining parts of the paper are as follows: Section 2 provides a theoretical overview of the DL algorithm (YOLOv4) used to extract cattle features from images. Section 3 illustrates the work methodology, including database preparation and brief work analysis. Section 4 reports the experimental results. It also includes quantitative evaluations and a comparative analysis of the obtained results. Finally, Section 5 provides the conclusions and future directions.

## 2. Framework for Cow Breed Detection

Real-time vision-based applications not only require accuracy but also demand fast detection with the ability to recognize a wide variety of objects. Although traditional object detection algorithms (like RCNN, fast RCNN, and faster RCNN) provide accurate detection, they are slower [29]. Therefore, to increase the detection speed, a single-shot detector (SSD) has been introduced, which can detect multiple objects at a significant rate of 22–59 frames per second (FPS) [30]. However, it exhibits poor accuracy in the detection of small objects. Unlike conventional CNN architectures, YOLOv4 can be easily used in real-time applications due to its fast and accurate detection. Therefore, YOLOv4 seems to be a perfect choice for object detection tasks. This algorithm is based on regression, i.e., instead of selecting regions of interest in an image, it predicts classes and bounding boxes for the whole image in one algorithm run. The parameters required to describe a bounding box are:

4. Bounding box's centre ( $b_x$  and  $b_y$ )
5. Width ( $b_w$ )
6. Height ( $b_h$ )
7. Class of an object ( $c$ ) (such as Marchigiana, White Park, etc.).

Along with the above-mentioned parameters, YOLOv4 also predicts the probability of containing an object ( $p_c$ ) in the bounding box as illustrated in Figure 2.



**Figure 2.** Parameters describing the bounding box.

The first edition of YOLO was YOLOv1 (with 24 convolutional layers), which was trained on the ImageNet-1000 dataset. It can detect objects with a speed of 45 FPS [31]. It outperforms conventional detection methods (like DPM and R-CNN) in terms of accuracy and speed. However, it has difficulty detecting small objects, mainly if they appear as a cluster. Therefore, another version of YOLO (known as YOLOv2) was introduced, which significantly improved the performance of object detection models. It offers the accuracy of faster R-CNN and the speed of SSD [32]. Due to the multi-scale training of the YOLOv2 network, it can detect and classify objects with different configurations and dimensions. Compared to its predecessor (i.e., YOLOv1), YOLOv2 can detect smaller objects more accurately. To make object detection algorithms more accurate and faster, YOLOv3 was launched, which accurately classifies objects in real-time applications [33]. For multi-label classification, it uses logistic classifiers instead of SoftMax. In 2020, YOLO evolved into YOLOv4, which uses YOLOv3 as its head with some changes in the backbone and neck [34]. It gives remarkable results, with a hike of 10% in accuracy and 12% in speed compared to YOLOv3. Therefore, in this study, YOLOv4 is used to detect cow breeds. The following subsections discuss the features and architecture of the YOLOv4 detection network.

### 2.1. YOLOv4

YOLOv4 has the edge over YOLOv3, as it implements a new architecture in the backbone, modifies the neck, and achieves a real-time speed of 65 FPS on Tesla V100. In addition, there is no need to use expensive GPUs for training i.e., training can be done on a single conventional GPU with great accuracy. YOLOv4 integrates special features within the bag of freebies and bag of specials as discussed below:

**Bag of freebies (BoF):** Accuracy is improved by changing the training strategy without increasing inference costs. To increase the robustness of images obtained from distinct environments, it uses data augmentation, which increases the variability of the input images. Furthermore, it solves the problem of photometric distortion by adjusting an images' brightness, hue, saturation, contrast, and noise. For geometric distortion, input images are randomly scaled, cropped, flipped, and rotated at some angle. In addition to data augmentation, BoF also solves object occlusion issues.

**Bag of specials (BoS):** It contains different post-processing modules that significantly enhance object detection accuracy at the cost of a slight rise in inference time. Figure 3 illustrates various methods present in BoS.

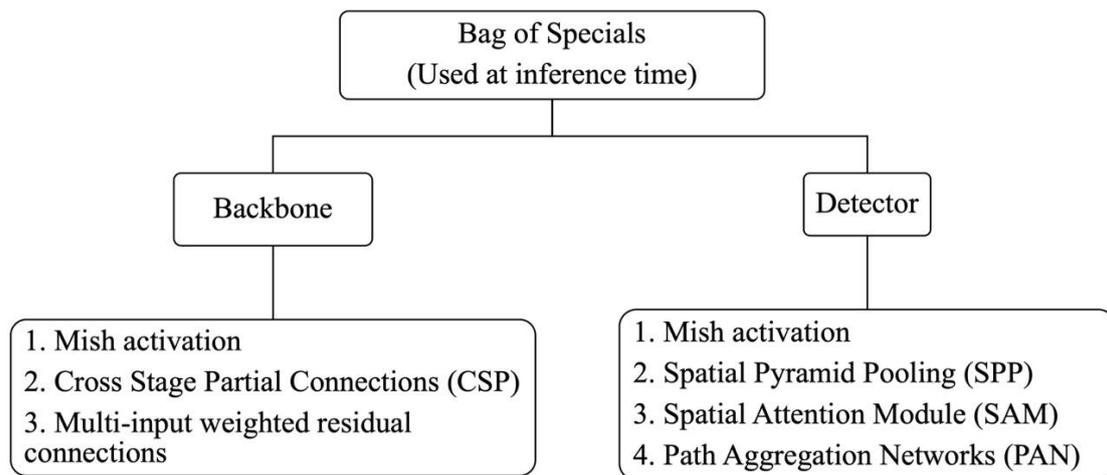


Figure 3. Different methods present in bag of specials.

2.2. YOLOv4 Architecture

As shown in Figure 4, the YOLOv4 architecture has three parent blocks: backbone, neck, and head (dense prediction).

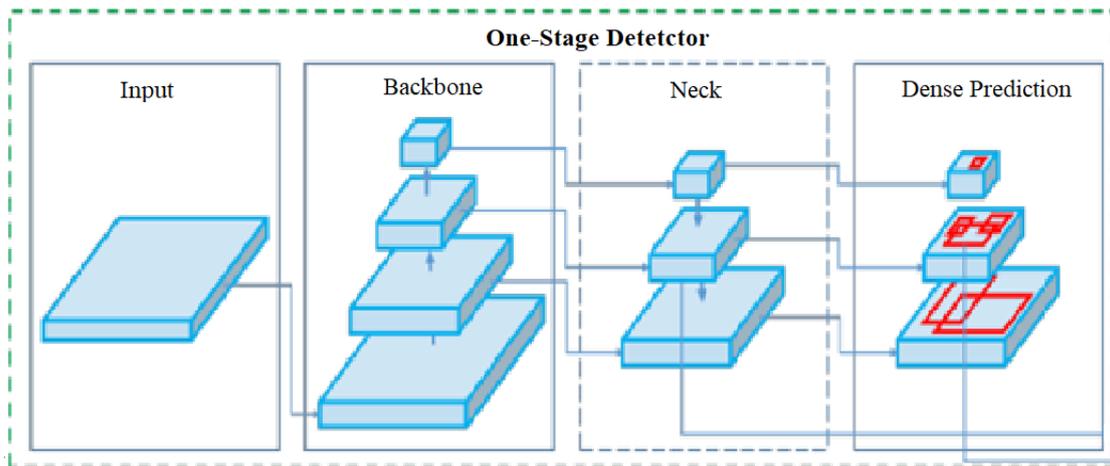


Figure 4. YOLOv4 architecture [34].

**Backbone:** The CSPDarknet53 network is used as a backbone to extract essential features from the input image. Cross-stage-partial net (CSPNet) divides the feature map of the base layer into two segments, as illustrated in Figure 5. A dense block contains multiple convolution layers that take the output of all the preceding layers and merge them with the current layer. DenseNet contains multiple dense blocks connected with transition layers (including convolution and pooling layers).

**Neck:** The neck’s main contribution to detection is combining feature maps from different stages. It enhances the information gathered from the backbone layer and feeds it into the head. It concatenates semantic-rich information (from the feature map of the top-down stream) with the spatial-rich information (from the bottom-up stream’s feature map) and feeds the concatenated output into the head.

**Head:** To perform dense prediction, YOLOv3 serves as the head of the YOLOv4 architecture. As a result, it provides the final prediction along with a vector containing the predicted coordinates of the bounding box and the associated confidence score with a class label.

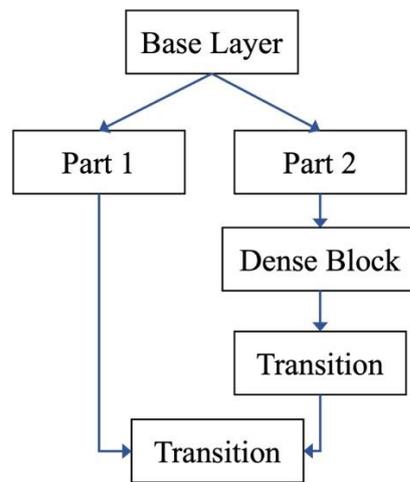


Figure 5. CSP DenseNet.

### 3. Methodology

Figure 6 illustrates the flow chart of the training process using the YOLOv4 algorithm. The first task in any DL algorithm is to prepare the dataset. For this purpose, 1835 images (which contain eight breeds of cows) are collected using web mining techniques. However, as DL models are data-driven, data augmentation has been performed on the acquired images to avoid the risk of overfitting. As shown in Figure 7, data augmentation involves a group of techniques that enhance the size of training datasets. An instance of augmented images is shown in Figure 8.

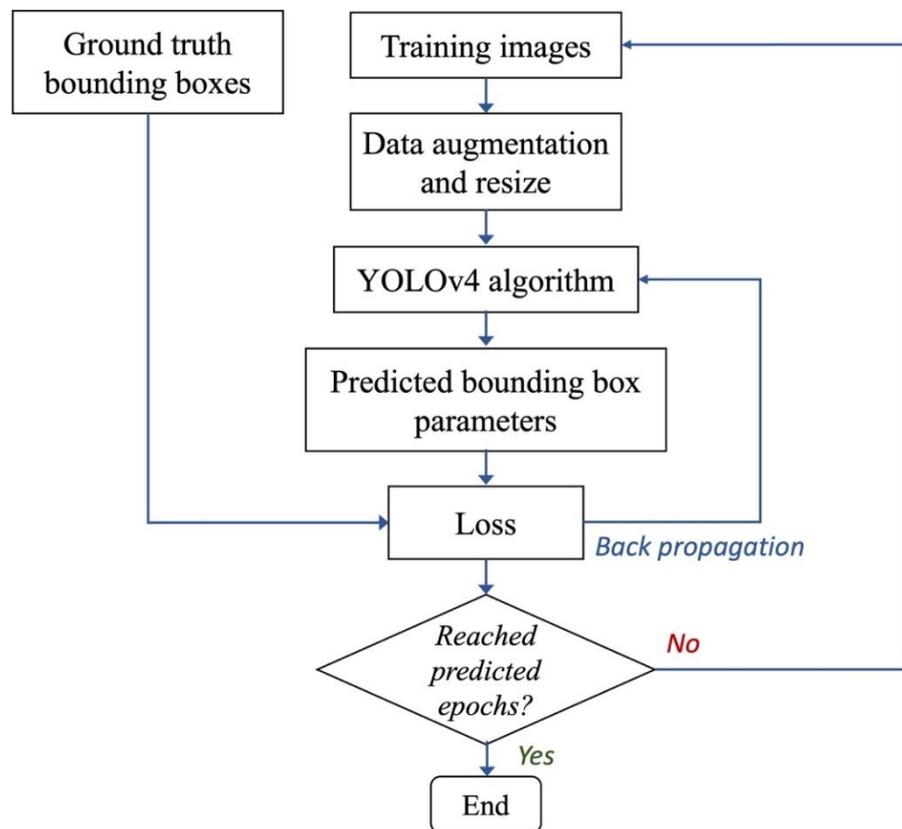


Figure 6. Training phase.

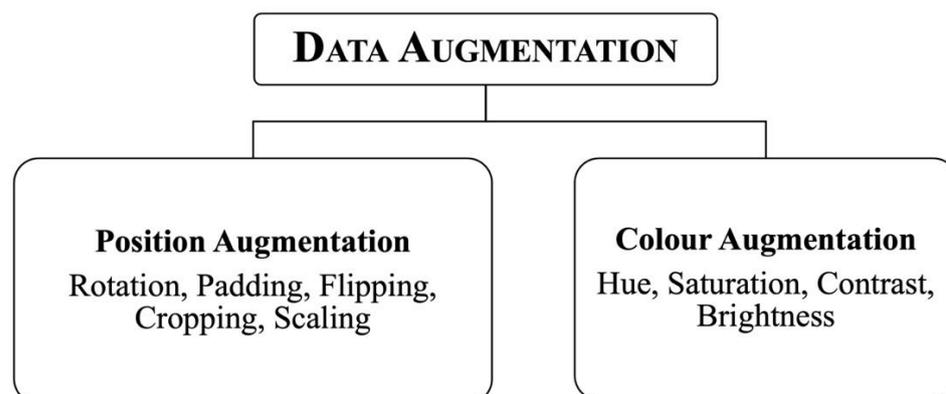


Figure 7. Data augmentation techniques.



Figure 8. Sample augmented images.

Further, the present work employs the transfer learning approach because a large dataset is required to train the model from scratch. Therefore, pre-trained weights (yolov4.conv.137) are applied as initial weights at the beginning of training. Moreover, the developed custom dataset of 1835 images has been randomly segmented into the subsequent phases: (1) training phase in which 90% of images (1662) are employed to train the proposed model, (2) validation phase containing 141 samples to validate the model, and (3) testing phase, which includes the remaining 32 images. The division of the dataset for each breed is illustrated in Table 2.

Table 2. Dataset division.

Classes	Training Images	Validation Images	Test Images
Afrikaner cattle	218	16	4
Brown Swiss cattle	203	16	4
Gyr cattle	202	14	4
Holstein Friesian	189	19	4
Limousin cattle	232	17	4
Marchigiana	163	18	4
Simmental cattle	221	22	4
White Park cattle	234	19	4

As discussed earlier, the network is trained by YOLOv4 for the detection of eight breeds of a cow (Afrikaner, Brown Swiss, Gyr, Holstein Friesian, Limousin, Marchigiana, White Park, and Simmental cattle). Moreover, the performance of the YOLOv4 algorithm is evaluated by training the model on different sets of training parameters. The parametric settings used to train the model via YOLOv4 are tabulated in Table 3. The whole investiga-

tion is performed on Nvidia RTX 2060 GPU, and the environment uses Visual Studio 2017 to compile the entire script.

**Table 3.** YOLOv4 configuration parameters.

Parameters	Different Sets of Training Parameters			
	416 × 416	416 × 416	608 × 608	608 × 608
Image size **	416 × 416	416 × 416	608 × 608	608 × 608
Channels	3	3	3	3
Momentum	0.949	0.949	0.949	0.949
Batch	64	64	64	64
Subdivisions *	64	64	64	64
Decay	0.0005	0.0005	0.0005	0.0005
Learning rate	0.0013	0.0013	0.0013	0.0013
max_batches *	20000	20000	20000	20000
Policy	Steps	Steps	Steps	Steps
Steps *	16,000, 18,000	16,000, 18,000	16,000, 18,000	16,000, 18,000
Scale	0.1, 0.1	0.1, 0.1	0.1, 0.1	0.1, 0.1
Classes *	8	8	8	8
Filters *	39	39	39	39
<i>IoU</i> _threshold **	0.50	0.75	0.50	0.75

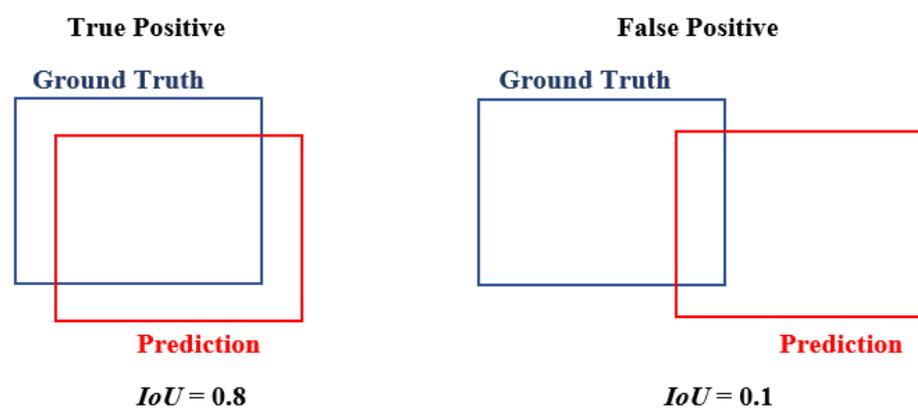
\* represents the parameters modified in the original YOLOv4 C.F.G. \*\* represents the parameters modified to evaluate the performance of YOLOv4. Note: filters = {Bounding box coordinates (5) + Total number of classes (8)} × Number of indices of anchors (3) = 39.

### 3.1. Evaluation Metrics

During training, intersection over union (*IoU*) is calculated by matching the detected bounding box with the ground truth box. It can be determined via Equation (1) [35].

$$IoU = \frac{\text{area of overlap between ground truth and detected bounding box}}{\text{area of union between the two boxes}} \quad (1)$$

Figure 9 illustrates an example of *IoU* computation. In this example, the *IoU*\_threshold has been taken as 0.5. If the prediction is greater than this threshold, it is classified a true positive; otherwise, detection is designated a false positive. Thus, by changing the *IoU*\_threshold, the model will give different true or false positives for the same prediction. The results are validated by computing the below-mentioned performance metrics:



**Figure 9.** Example of true and false positives.

### 3.1.1. Precision–Recall (PR) Curve

PR curves summarize the trade-off between *precision* and *recall* values. It is plotted at different probability thresholds representing *precision* along the *y*-axis and *recall* along the *x*-axis.

*Precision*: The detected bounding box is compared with the ground-truth box and describes how good a model is at predicting the positive class. It is represented by Equation (2) [12].

$$Precision = \frac{\text{total no. of objects detected correctly}}{\text{total objects detected}} = \frac{N_{TP}}{N_{TP} + N_{FP}} \quad (2)$$

here,  $N_{TP}$  = number of predictions that resembles the ground-truth boxes (true positive)

$N_{FP}$  = number of false detections (false positives)

*Recall*: Recall denotes the sensitivity, i.e., how many positive predictions are captured from total ground-truth boxes. It is generally expressed by Equation (3) [36].

$$Recall = \frac{\text{total no. of objects detected correctly}}{\text{no. of ground truth objects}} = \frac{N_{TP}}{N_{TP} + N_{FN}} \quad (3)$$

here,  $N_{FN}$  = number of ground-truth objects that could not be detected (false negatives)

A model with perfect skill is depicted as a point at (1, 1) where both *precision* and *recall* values are high. Therefore, the accuracy of the model increases as it moves toward point (1, 1). The area under the PR curve is known as average precision (AP), and the mean of APs for all classes is termed as the mean average precision (mAP). These are represented by Equations (4) and (5), respectively [37].

$$AP = \int_0^1 p(r) dr \quad (4)$$

$$mAP = \frac{\sum_{i=1}^N AP_i}{N} \quad (5)$$

Precision and recall are encapsulated in another well-known evaluation metric, the  $F_1$  score. It is the harmonic mean of *precision* and *recall* and computed by Equation (6) [38].

$$F_1 \text{ score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (6)$$

### 3.1.2. Confusion Matrix

The ratio of true positives to total predictions made determines the Classification accuracy. It can be misleading if the data has more than two classes or does not have a balanced dataset. For example, if a classification accuracy of 90% is obtained, it does not mean that all classes are being predicted equally. There is a probability that the model neglects one or two categories. Good accuracy can be achieved by predicting the most common class value, i.e., the class with the maximum number of training images. So, to visualize the model's performance, a confusion matrix is employed. It summarizes prediction results with the number of correct and incorrect predictions encapsulated class-wise in a matrix, as shown in Figure 10 [39].

		Truth Data	
		Positive	Negative
Classifier Results	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Figure 10. Confusion matrix.

A confusion matrix is not only limited to true/false positives but also helps in estimating the performance by calculating other evaluation metrics including accuracy and *kappa*.

Overall accuracy (OA): The perfect classification is represented with 100% accuracy where all of the classes were classified correctly. It is calculated by Equation (7):

$$OA = \frac{\text{total no. of objects detected correctly}}{\text{total objects detected}} \quad (7)$$

The diagonal elements in the matrix represent the number of predictions classified correctly, and the total number of values represents the total objects detected. OA is the easiest to analyze from the confusion matrix. However, it only provides basic accuracy information as the class-wise evaluation is missing. Therefore, class-wise *precision*, *recall*, and Cohen's *kappa* are calculated from the matrix, as discussed below:

*Precision and Recall*: The confusion matrix helps in estimating *precision* and *recall* for each class. The precision and recall values are calculated using Equations (2) and (3).

*Cohen's Kappa*: Another metric calculated from the confusion matrix is *kappa*. It measures the agreement between classification and truth values, i.e., it calculates inter-rater reliability. It is generally a more robust measurement than simple accuracy computation, as *kappa* considers the possibility of the agreement occurring by chance (false positives). Mathematically, it is computed by Equation (8) [40].

$$kappa = \frac{N \sum_{i=1}^n m_{i,i} - \sum_{i=1}^n (G_i C_i)}{N^2 - \sum_{i=1}^n (G_i C_i)} \quad (8)$$

here,  $n$ : total number of classes,  $N$ : total number of classified values compared to truth values,  $m_{i,i}$ : values found along the diagonal of the confusion matrix,  $C_i$ : total number of predictions belonging to class  $i$ ,  $G_i$ : total number of truth values belonging to class  $i$ .

It varies in the range (0, 1). The lower bound shows no agreement, whereas the upper bound indicates perfect agreement. Practically, Cohen's *kappa* removes the possibility of random-guess agreeing.

#### 4. Experimental Results and Discussion

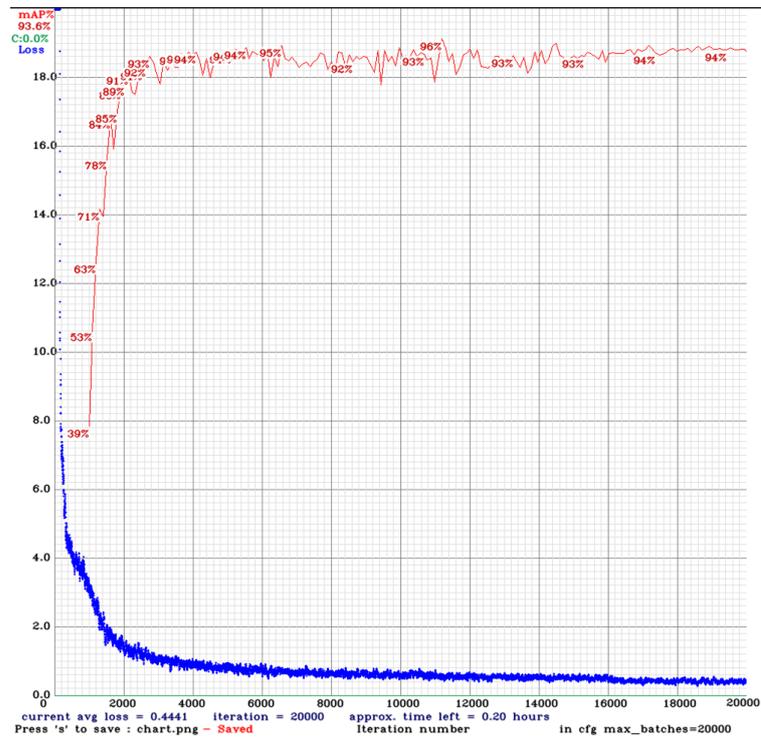
Training any DL model from scratch requires substantial data and computational resources. Therefore, the transfer learning approach is employed to develop the proposed cow breed detection model using pre-trained weights of YOLOv4. This helps the model to learn better attributes, resulting in improved detection capability. Further, the model is trained for 20,000 iterations with the hyperparameters mentioned in Table 3. The training graph of the model with the developed custom dataset is mapped in Figure 11.

The corresponding performance parameters are provided in Table 4. Training time increases when the image size (i.e., width  $\times$  height) increases. It indicates that YOLOv4 runs faster for smaller images. From the graph illustrated in Figure 11, it is spotted that the *mAP* value increases gradually when the *IoU*\_threshold is 0.75 as compared to 0.50. By changing the *IoU*\_threshold, the model will give different TRUE or FALSE positives for the same prediction, consequently affecting the *mAP* values.

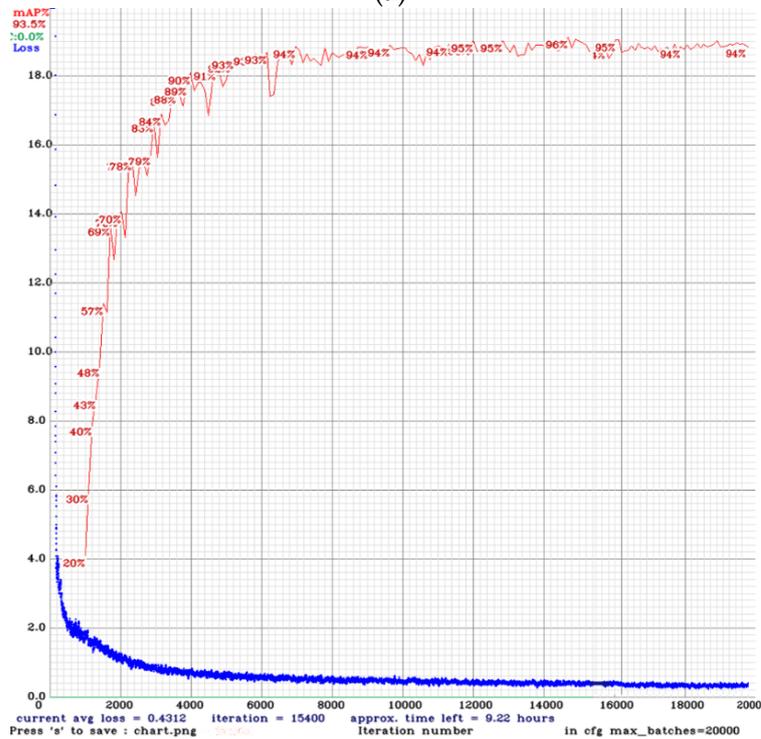
Figure 12 shows the statistical analysis of performance metrics obtained from the last epoch. Table 4 and Figure 12 show that recall is higher (almost 2%) when the *IoU*\_threshold is low, as more predictions turn to positives at a low threshold. However, *mAP* increases by 0.5% for an *IoU*\_threshold of 0.75 compared to an *IoU*\_threshold of 0.5. A probable reason might be the decrease in false positives with an increase in the threshold.

The dataset contains only 32 test images that generally have only one type of breed. Therefore, to explore the efficiency of the model under complex circumstances, a collage is developed containing images of cows from multiple breeds. The detection results on sample test images along with collage are shown in Figure 13, and its comparative analysis is demonstrated in Table 5. It is observed that the model gives high *precision* (>2%) when

image resolution increases. It is also noticed that detection time increases by 65% with an increase in image size as YOLOv4 runs slow on large-sized images.

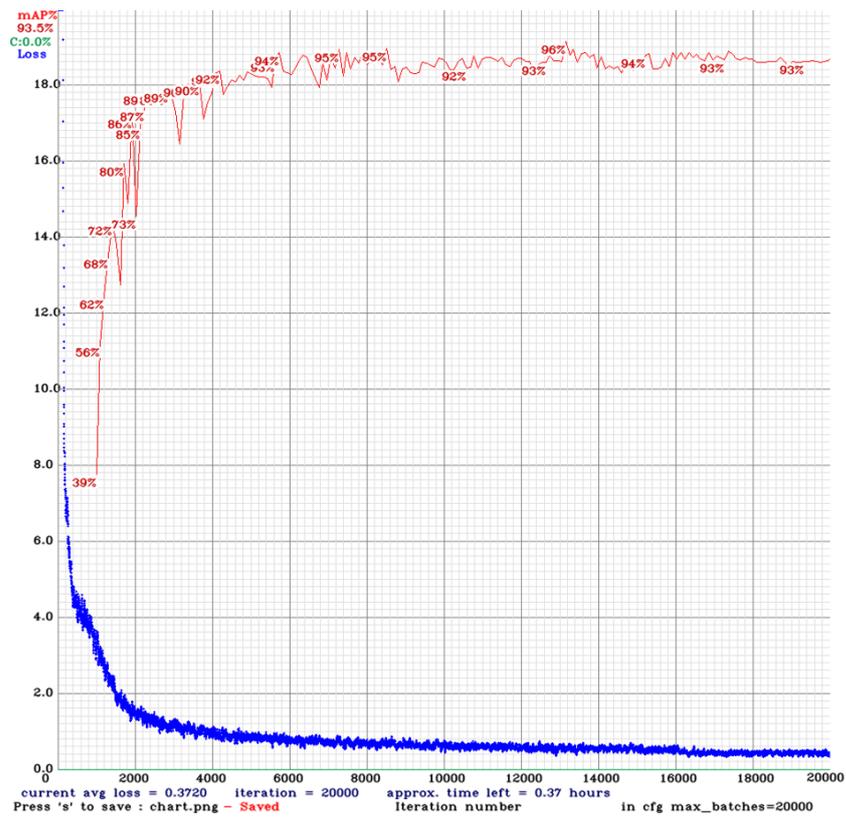


(a)

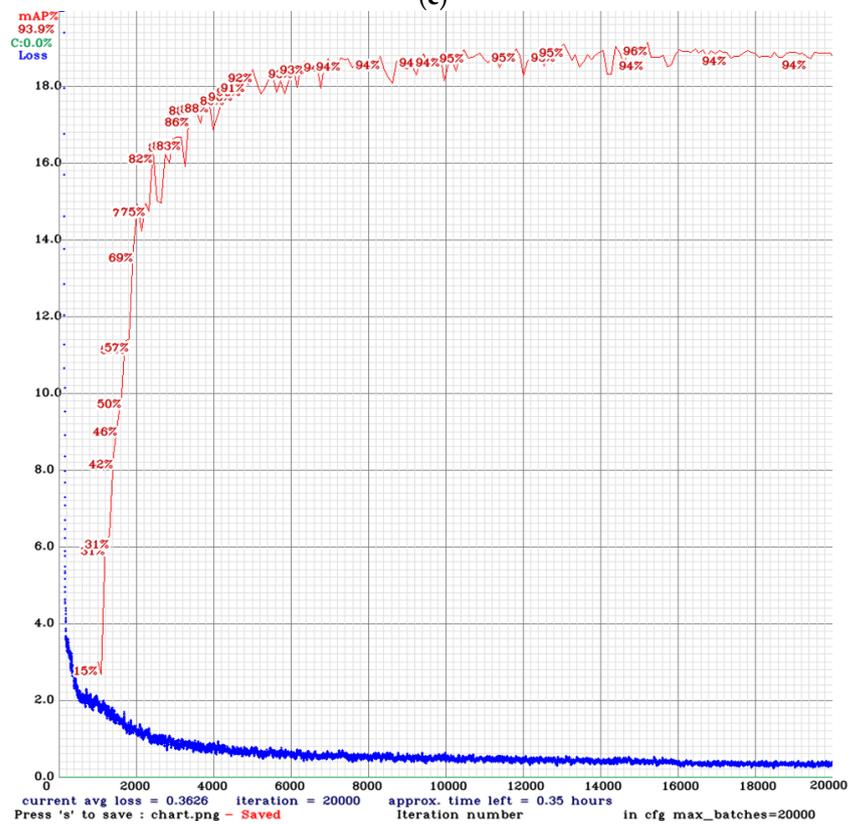


(b)

Figure 11. Cont.



(c)

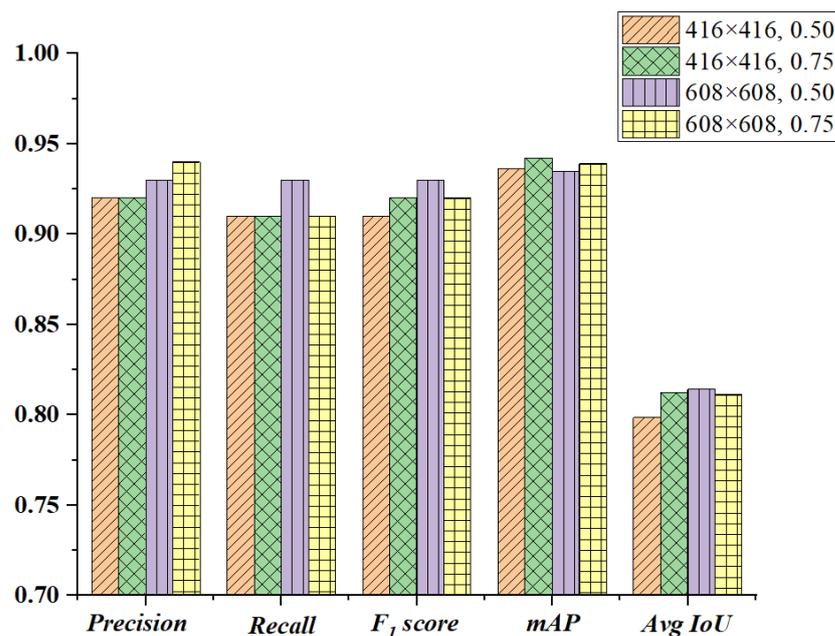


(d)

Figure 11. Graph of YOLOv4 trained with network size and *IoU*\_threshold of (a)  $416 \times 416$ , 0.50 (b)  $416 \times 416$ , 0.75 (c)  $608 \times 608$ , 0.50 (d)  $608 \times 608$ , 0.75.

**Table 4.** Performance parameters at different iterations of training.

No. of Iterations	Training Parameters	Precision		Recall		F <sub>1</sub> Score		mAP%		Avg IoU%	
		0.50	0.75	0.50	0.75	0.50	0.75	0.50	0.75	0.50	0.75
1000	416 × 416	0.48	0.31	0.32	0.07	0.39	0.12	39.44	21.00	35.63	20.76
	608 × 608	0.44	0.80	0.15	0.02	0.23	0.03	38.55	15.25	31.72	57.91
3000	416 × 416	0.87	0.81	0.89	0.71	0.88	0.76	92.12	80.20	70.66	64.58
	608 × 608	0.86	0.83	0.81	0.69	0.83	0.75	86.04	74.18	68.97	65.06
5000	416 × 416	0.90	0.89	0.84	0.85	0.87	0.87	91.34	88.87	75.23	74.70
	608 × 608	0.91	0.89	0.88	0.86	0.90	0.88	90.09	90.84	75.65	73.54
7000	416 × 416	0.93	0.91	0.89	0.90	0.91	0.90	92.95	93.42	78.27	73.58
	608 × 608	0.90	0.91	0.88	0.90	0.89	0.91	92.25	91.96	73.95	75.70
9000	416 × 416	0.93	0.90	0.89	0.88	0.91	0.89	93.06	93.88	77.80	75.90
	608 × 608	0.92	0.93	0.92	0.92	0.92	0.93	92.80	94.33	78.03	76.71
11,000	416 × 416	0.92	0.92	0.86	0.93	0.89	0.92	92.10	94.51	78.50	78.37
	608 × 608	0.89	0.94	0.91	0.91	0.90	0.93	93.83	92.78	76.22	78.48
13,000	416 × 416	0.91	0.92	0.91	0.91	0.91	0.91	93.16	94.57	78.08	78.31
	608 × 608	0.91	0.95	0.93	0.89	0.92	0.92	93.09	93.82	77.24	79.71
15,000	416 × 416	0.85	0.93	0.90	0.93	0.90	0.93	92.49	95.24	73.80	78.47
	608 × 608	0.90	0.93	0.92	0.92	0.91	0.92	92.99	95.05	77.07	78.75
17,000	416 × 416	0.93	0.93	0.93	0.93	0.93	0.93	94.64	94.35	80.96	80.62
	608 × 608	0.93	0.93	0.93	0.91	0.93	0.92	94.11	93.84	81.22	80.89
19,000	416 × 416	0.92	0.93	0.92	0.93	0.92	0.93	93.93	94.65	80.23	81.00
	608 × 608	0.93	0.94	0.92	0.92	0.93	0.93	93.03	94.37	81.35	81.29
20,000	416 × 416	0.92	0.92	0.91	0.92	0.91	0.92	93.64	94.22	79.85	81.24
	608 × 608	0.93	0.94	0.93	0.91	0.93	0.92	93.47	93.91	81.41	81.12



**Figure 12.** Comparison between different evaluation metrics.



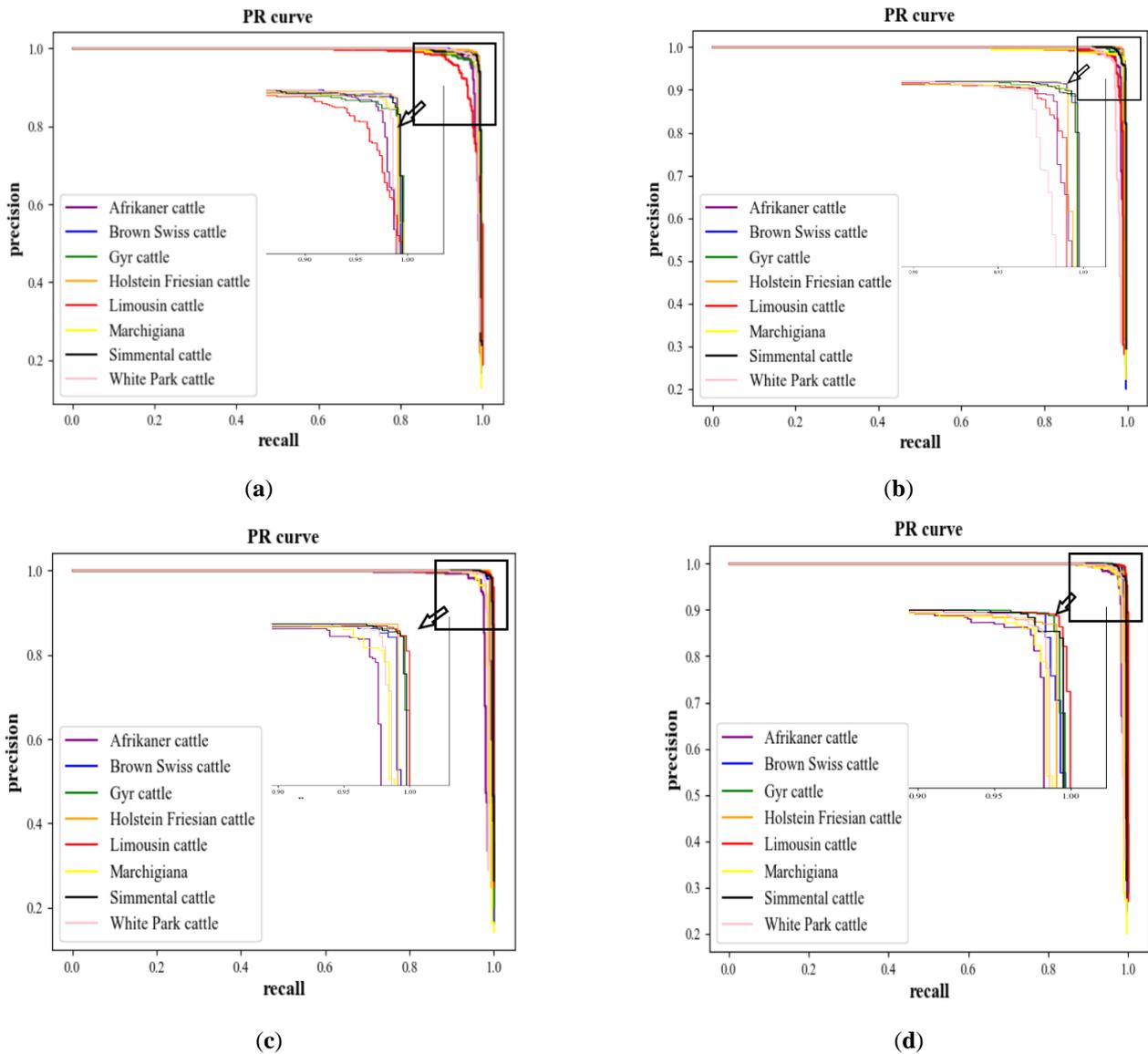
Figure 13. Sample of test images.

Table 5. Sample comparison based on detection.

Test Image	Total No. of Objects	No. of Positive Detections/Total No. of Detection				Predicted in Time (ms)			
		416 × 416		608 × 608		416 × 416		608 × 608	
		0.50	0.75	0.50	0.75	0.50	0.75	0.50	0.75
1	12	12/12	11/11	11/11	12/12	30.43	31.74	51.58	51.74
2	10	8/9	8/10	10/10	10/10	31.60	31.84	51.96	51.74
3	4	2/4	2/3	3/4	3/3	30.62	30.25	51.465	51.463
4	6	6/6	6/7	6/6	6/6	31.60	31.4	50.95	51.73
5	7	7/7	5/7	5/6	4/6	31.28	31.30	52.09	52.11
6	7	7/7	7/7	7/7	6/6	31.47	30.15	51.91	51.47
		Average precision				Average prediction time (ms)			
		93.30	86.60	95.40	95.30	31.17	31.11	51.66	51.71

### 4.1. PR Curve

Figure 14 shows the PR curves for different cases of network size and *IoU*\_threshold. From Figure 14, it is observed that the curves are closer to (1, 1) for each class for the fourth case (608 × 608, 0.75) compared to other ones. Furthermore, *mAP* is calculated from PR curves, which are greater in ‘608 × 608, 0.75’ by 0.17%, 0.15%, and 0.01% compared to ‘416 × 416, 0.50’, ‘416 × 416, 0.75’, and ‘608 × 608, 0.50’ respectively. This indicates that the PR curve shows visible change when the image resolution changes.



**Figure 14.** PR curves acquired with network size and *IoU*\_threshold (a) 416 × 416, 0.50 (b) 416 × 416, 0.75 (c) 608 × 608, 0.50 (d) 608 × 608, 0.75.

### 4.2. Confusion Matrix

As discussed earlier, the image resolution and *IoU*\_threshold significantly influence the performance of the object detection model. This is also reflected in the confusion matrix, as illustrated in Figure 15. In this study, predictions were taken as positive if the confidence score was greater than 1% to compute the confusion matrix. Since cows of studied classes share almost similar traits, it is very difficult for the model to extract features with lower threshold and image resolution. Hence, a large amount of misidentification and misclassification have been witnessed, particularly with 416 × 416, 0.50. However,

fine features may be recognized as image resolution improves, reducing the frequency of erroneous detection.

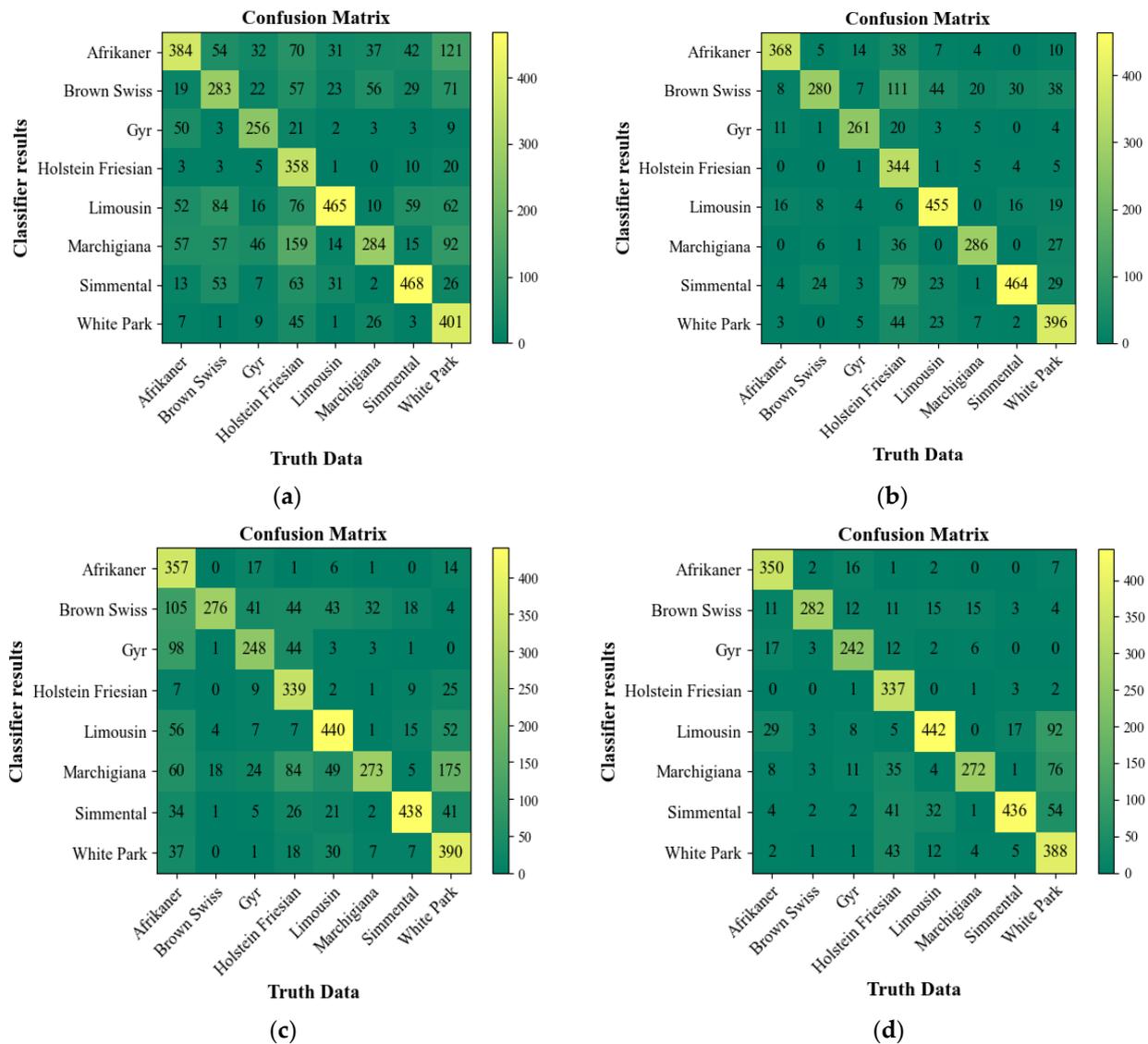


Figure 15. Confusion matrix obtained with network size and *IoU*\_threshold (a) 416 × 416, 0.50 (b) 416 × 416, 0.75 (c) 608 × 608, 0.50 (d) 608 × 608, 0.75.

*OA* calculated from confusion matrices is illustrated in Figure 16. It is observed that *OA* is greater in the fourth case (i.e., ‘608 × 608, 0.75’) by 33.27%, 3.28%, and 19.70% than ‘416 × 416, 0.50’, ‘416 × 416, 0.75’, and ‘608 × 608, 0.50’ respectively. This also validates the hypothesis that the performance of the model improves when the image resolution and threshold increase.

To further validate the model, class-wise *precision* and *recall* values are computed. The results are presented in Figures 17 and 18, respectively. The *precision* and *recall* values are small for an *IoU*\_threshold of 0.50 compared to 0.75 as the total objects detected are lower in the latter (reflected from confusion matrices in Figure 15).

Due to diversity in training image size, *precision* and *recall* follow the random trend when the images are resized from 416 × 416 to 608 × 608. The model performance is affected because the model uses a zero padding technique to fit the input image into the required image size.

Figure 19 also supports that the detection accuracy increases when both image resolution and threshold increases. *Kappa* is found to be greater in the fourth case (608 × 608,

0.75) with a hike of 42.10%, 3.84%, and 23.60% compared to '416 × 416, 0.50', '416 × 416, 0.75', and '608 × 608, 0.50' respectively.

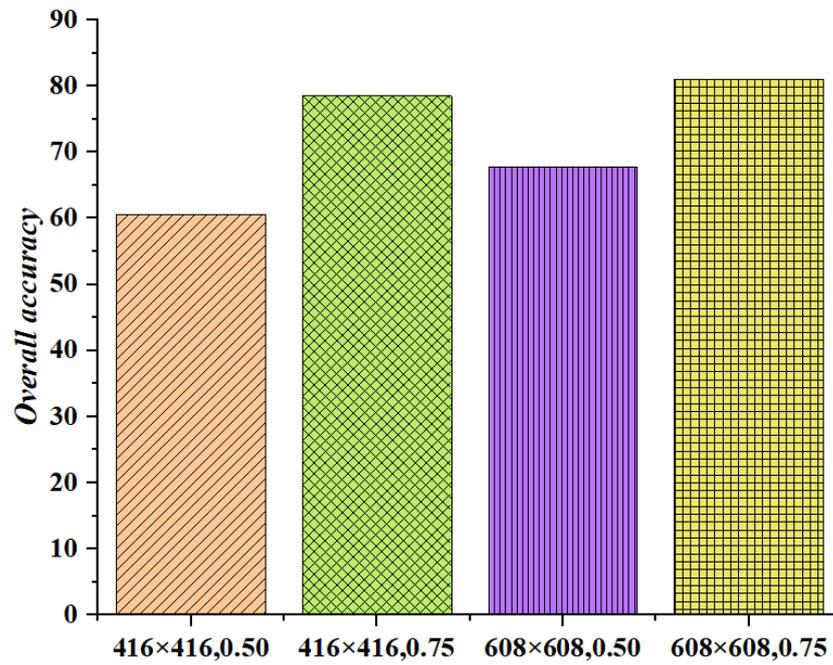


Figure 16. Overall accuracy (OA).

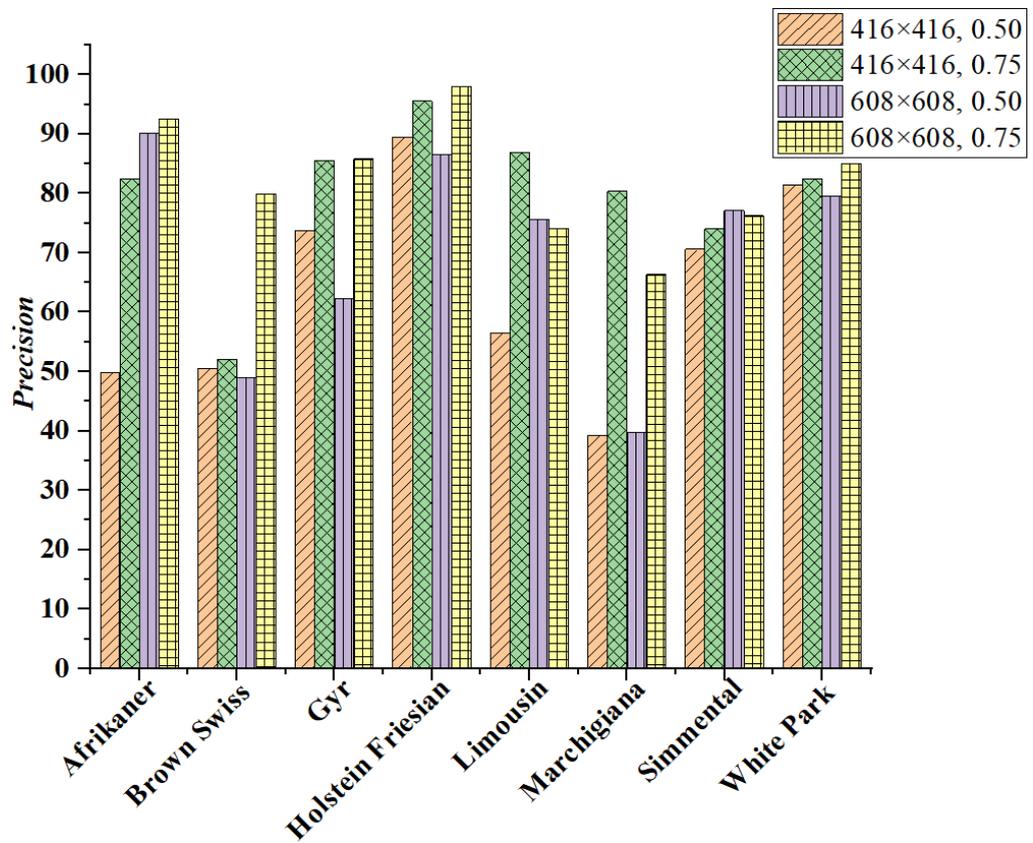


Figure 17. Class-wise precision.

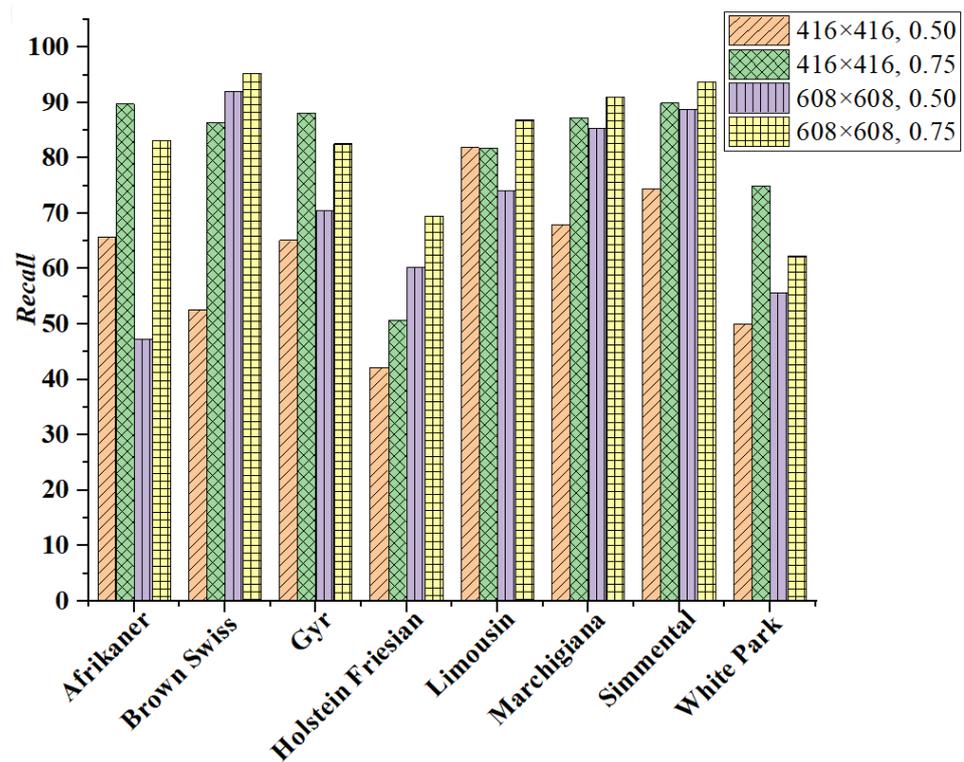


Figure 18. Class-wise recall.

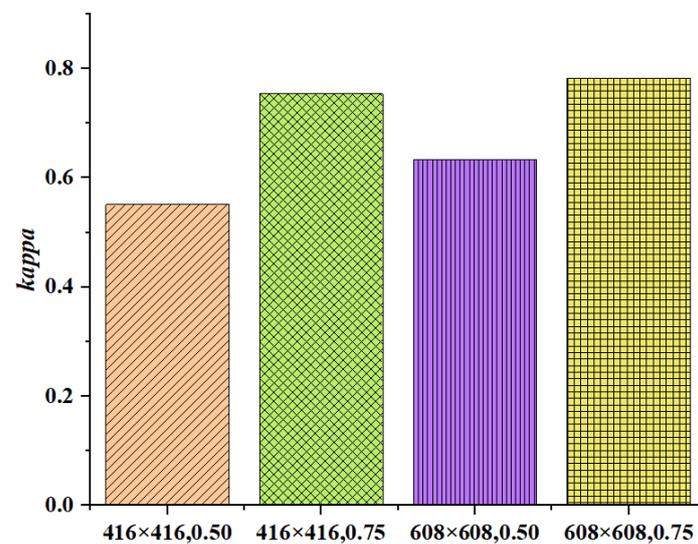


Figure 19. Kappa values quantified from the confusion matrix.

#### 4.3. Comparison with State-of-the-Art Models

The results above demonstrate the improved detection accuracy of the proposed model with  $608 \times 608$  image resolution and an *IoU*\_threshold of 0.75. Further, it is observed that most of the reported work detects only one breed. Therefore, the developed custom dataset is employed to train three other popular detection techniques (faster RCNN, SSD, and YOLOv3) for a fair comparison. Table 6 presents the class-wise accuracy of the proposed model relative to faster RCNN, SSD, and YOLOv3. This table also compares the speed of detection by these models.

**Table 6.** Comparative analysis.

S. No.	Base Model	Class								FPS
		A	BS	G	HF	L	M	S	WP	
1.	Faster RCNN [41]	85.21	78.36	73.87	84.76	77.84	76.19	74.13	80.17	3.04
2.	SSD [30]	73.48	69.14	66.25	72.83	69.27	69.27	71.06	75.43	4.60
3.	YOLOv3 [33]	84.67	77.55	71.39	83.92	74.13	72.04	72.57	79.84	9.22
4.	YOLOv4 (proposed)	87.59	81.11	82.44	88.37	78.19	76.88	77.46	82.32	19.61

A: Afrikaner, BS: Brown Swiss, G: Gyr, HF: Holstein Friesian, L: Limousin, M: Marchigiana, S: Simmental, WP: White Park.

It was computed that the model with YOLOv4 outclasses faster RCNN by a minimum and maximum margin of 0.45% for Limousin breed and 11.60% for Gyr class, respectively. Similarly, YOLOv4 dominates the detection accuracy of SSD by a minimum margin of 9.00% for all the categories. Similar trends were witnessed while comparing the performance of YOLOv4 with YOLOv3, as the lowest boost (3.11%) was observed for White Park. Moreover, the model developed with YOLOv4 significantly uplifted the inference speed.

To conclude, the experiments and results reported in this section validate the credibility of this work and place the proposed model among the top-ranked cow breed detection models. The methodology proposed in this work could be used in real-time scenarios to detect cow breeds, thereby assisting in the improvement of automatic livestock farming.

## 5. Conclusions

This work proposes a vision-based model to recognize the breed of a cow. YOLOv4, a DL algorithm, is applied to learn a discriminatory feature of cows with a limited training dataset. For this, a custom dataset for eight breeds of cow (Afrikaner, Brown Swiss, Gyr, Holstein Friesian, Limousin, Marchigiana, White Park, and Simmental cattle) was generated. To test the efficiency of the algorithm, PR curves and confusion matrix were drawn on test images, which demonstrates that the YOLOv4 algorithm works better with an image size of  $608 \times 608$  and *IoU*\_threshold of 0.75. Furthermore, *mAP* calculated from PR curves improve by 0.17%, 0.15%, and 0.01% in training image size and *IoU*\_thresholds of '416 × 416, 0.50', '416 × 416, 0.75' and '608 × 608, 0.50', respectively. Consequently, the PR curve shows visible changes, with variations in image resolution. Overall accuracy, *OA*, calculated from the confusion matrix is more in '608 × 608, 0.75' by 33.27%, 3.28%, and 19.70% than '416 × 416, 0.50', '416 × 416, 0.75', and '608 × 608, 0.50', respectively. Another metric, i.e., *kappa*, also indicates that the model performs better when the image size and the *IoU*\_threshold are '608 × 608, 0.75'. In '608 × 608, 0.75', *Kappa* increases by 42.1%, 3.84% and 23.60% than '416 × 416, 0.50', '416 × 416, 0.75', and '608 × 608, 0.50', respectively. Overall, the experimental results demonstrate that the model accuracy can be improved by training YOLOv4 on images with high resolution, with a greater *IoU*\_threshold. Further, the developed cow breed model is compared with the models developed by employing faster RCNN, SSD, and YOLOv3. The comparative analysis validates the improved performance of the cow breed detection model with YOLOv4.

Further research will focus on video tracking for effective identification via surveillance. As the present work on the individual identification of the cow breed (through images) yields highly accurate results, it would be interesting to incorporate simple tracking techniques between video frames to check the efficiency of this work, more precisely in case of the heavy bunching of cows. Further, the proposed methodology needs to be trained on aerial-view-based datasets with multiple breeds to enhance the accuracy and robustness of the model. This will be addressed in future works. In addition, the scalability of our approach to large populations remains to be tested. This will open new doors to deploy vision-based algorithms for the precision livestock farming sector.

**Author Contributions:** Conceptualization, H.G. and P.J.; Data curation, H.G. and R.K.A.; Formal analysis, P.J.; Funding acquisition, N.F.S.; Investigation, H.G., O.P.V., A.A.A. and V.M.; Methodology, H.G., P.J., O.P.V. and R.K.A.; Software, H.G. and A.A.A.; Supervision, N.F.S. and V.M.; Visualization, P.J. and V.M.; Writing—original draft, H.G., P.J., O.P.V., R.K.A. and V.M.; Writing—review & editing, A.A.A. and N.F.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R66), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

**Data Availability Statement:** The dataset generated and analyzed during the current work is available upon reasonable request from the corresponding author.

**Acknowledgments:** The authors express their gratitude to Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R66), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors are also thankful to Ganpate J. Dahe, Hydromer Inc., USA for his valuable suggestions.

**Conflicts of Interest:** The authors declare no conflict of interest.

### Abbreviations

AP	Average precision
BoF	Bag of freebies
BoS	Bag of specials
CNN	Convolutional neural network
CSPNet	Cross-stage-partial net
DL	Deep learning
FPS	Frames per second
GDP	Gross domestic product
GPU	Graphics processing unit
GVA	Gross value added
IoU	Intersection over union
LSTM	Long short-term memory
mAP	Mean average precision
OA	Overall accuracy
PR	Precision–recall
RCN	Recurrent convolutional network
RCNN	Region-based convolutional neural network
RFID	Radio frequency identification
SSD	Single-shot detector
YOLO	You only look once

### References

1. The World Bank. Agriculture Overview: Development News, Research, Data | World Bank. 2022. Available online: <https://www.worldbank.org/en/topic/agriculture/overview#1> (accessed on 18 April 2022).
2. Ministry of Statistics & Programme Implementation, Government of India. Provisional Estimates of Annual National Income, 2020–2021 and Quarterly Estimates (Q4) of Gross Domestic Product, 2020–2021. Available online: <https://pib.gov.in/pressreleasepage.aspx?prid=1723153> (accessed on 18 April 2022).
3. Belhekar, S. Role of Dairy Industry in Rural Development in India. *Indian J. Res* **2016**, *5*, 509–510.
4. Singh, R. Doubling Farmer’s Income: The Case of India. *World Food Policy* **2019**, *5*, 24–34. [[CrossRef](#)]
5. Kelsey, J.A.; Corl, B.A.; Collier, R.J.; Bauman, D.E. The Effect of Breed, Parity, and Stage of Lactation on Conjugated Linoleic Acid (CLA) in Milk Fat from Dairy Cows. *J. Dairy Sci.* **2003**, *86*, 2588–2597. [[CrossRef](#)]
6. Penasa, M.; Tiezzi, F.; Sturaro, A.; Cassandro, M.; De Marchi, M. A Comparison of the Predicted Coagulation Characteristics and Composition of Milk from Multi-Breed Herds of Holstein-Friesian, Brown Swiss and Simmental Cows. *Int. Dairy J.* **2014**, *35*, 6–10. [[CrossRef](#)]
7. Thompkinson, D.K. *Quality Milk Production and Processing Technology*; New India Publishing Agency: Delhi, India, 2012.
8. Awad, A.I. From Classical Methods to Animal Biometrics: A Review on Cattle Identification and Tracking. *Comput. Electron. Agric.* **2016**, *123*, 423–435. [[CrossRef](#)]
9. Hossain, M.E.; Kabir, M.A.; Zheng, L.; Swain, D.L.; McGrath, S.; Medway, J. A Systematic Review of Machine Learning Techniques for Cattle Identification: Datasets, Methods and Future Directions. *Artif. Intell. Agric.* **2022**, *6*, 138–155. [[CrossRef](#)]

10. Shojaeipour, A.; Falzon, G.; Kwan, P.; Hadavi, N.; Cowley, F.C.; Paul, D. Automated Muzzle Detection and Biometric Identification via Few-Shot Deep Transfer Learning of Mixed Breed Cattle. *Agronomy* **2021**, *11*, 2365. [[CrossRef](#)]
11. Gupta, H.; Verma, O.P. Monitoring and Surveillance of Urban Road Traffic Using Low Altitude Drone Images: A Deep Learning Approach. *Multimed. Tools Appl.* **2021**, *81*, 19683–19703. [[CrossRef](#)]
12. Kumar, S.; Gupta, H.; Yadav, D.; Ansari, I.A.; Verma, O.P. YOLOv4 Algorithm for the Real-Time Detection of Fire and Personal Protective Equipments at Construction Sites. *Multimed. Tools Appl.* **2021**, *81*, 22163–22183. [[CrossRef](#)]
13. Abu Jwade, S.; Guzzomi, A.; Mian, A. On Farm Automatic Sheep Breed Classification Using Deep Learning. *Comput. Electron. Agric.* **2019**, *167*, 105055. [[CrossRef](#)]
14. Borwarginn, P.; Thongkanchorn, K.; Kanchanapreechakorn, S.; Kusakunniran, W. Breakthrough Conventional Based Approach for Dog Breed Classification Using CNN with Transfer Learning. In Proceedings of the 2019 11th International Conference on Information Technology and Electrical Engineering (ICITEE), Pattaya, Thailand, 10–11 October 2019; IEEE: Piscataway, NJ, USA, 2019. [[CrossRef](#)]
15. Niemi, J.; Tantt, J.T. Deep Learning Case Study for Automatic Bird Identification. *Appl. Sci.* **2018**, *8*, 2089. [[CrossRef](#)]
16. Weber, F.d.L.; Weber, V.A.d.M.; Menezes, G.V.; Oliveira, A.d.S., Jr.; Alves, D.A.; de Oliveira, M.V.M.; Matsubara, E.T.; Pistori, H.; de Abreu, U.G.P. Recognition of Pantaneira Cattle Breed Using Computer Vision and Convolutional Neural Networks. *Comput. Electron. Agric.* **2020**, *175*, 105548. [[CrossRef](#)]
17. Awad, A.I.; Hassaballah, M. Bag-of-Visual-Words for Cattle Identification from Muzzle Print Images. *Appl. Sci.* **2019**, *9*, 4914. [[CrossRef](#)]
18. Kumar, S.; Pandey, A.; Sai Ram Satwik, K.; Kumar, S.; Singh, S.K.; Singh, A.K.; Mohan, A. Deep Learning Framework for Recognition of Cattle Using Muzzle Point Image Pattern. *Measurement* **2018**, *116*, 1–17. [[CrossRef](#)]
19. Kumar, S.; Singh, S.K.; Singh, R.S.; Singh, A.K.; Tiwari, S. Real-Time Recognition of Cattle Using Animal Biometrics. *J. Real-Time Image Process.* **2017**, *13*, 505–526. [[CrossRef](#)]
20. Bello, R.W.; Talib, A.Z.H.; Mohamed, A.S.A. Bin Deep Learning-Based Architectures for Recognition of Cow Using Cow Nose Image Pattern. *Gazi Univ. J. Sci.* **2020**, *33*, 831–844. [[CrossRef](#)]
21. Lu, Y.; He, X.; Wen, Y.; Wang, P.S.P. A New Cow Identification System Based on Iris Analysis and Recognition. *Int. J. Biom.* **2014**, *6*, 18–32. [[CrossRef](#)]
22. Kumar, S.; Singh, S.K. Cattle Recognition: A New Frontier in Visual Animal Biometrics Research. *Proc. Natl. Acad. Sci. India Sect. A Phys. Sci.* **2019**, *90*, 689–708. [[CrossRef](#)]
23. Manoj, S.; Rakshith, S.; Kanchana, V. Identification of Cattle Breed Using the Convolutional Neural Network. In Proceedings of the 2021 3rd International Conference on Signal Processing and Communication ICPSC, Coimbatore, India, 13–14 May 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 503–507. [[CrossRef](#)]
24. Qiao, Y.; Kong, H.; Clark, C.; Lomax, S.; Su, D.; Eiffert, S.; Sukkarieh, S. Intelligent Perception-Based Cattle Lameness Detection and Behaviour Recognition: A Review. *Animals* **2021**, *11*, 3033. [[CrossRef](#)]
25. Andrew, W.; Greatwood, C.; Burghardt, T. Visual Localisation and Individual Identification of Holstein Friesian Cattle via Deep Learning. In Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops ICCVW, Venice, Italy, 22–29 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 2850–2859. [[CrossRef](#)]
26. Andrew, W.; Greatwood, C.; Burghardt, T. Aerial Animal Biometrics: Individual Friesian Cattle Recovery and Visual Identification via an Autonomous UAV with Onboard Deep Inference. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 237–243. [[CrossRef](#)]
27. Tassinari, P.; Bovo, M.; Benni, S.; Franzoni, S.; Poggi, M.; Mammi, L.M.E.; Mattoccia, S.; Di Stefano, L.; Bonora, F.; Barbaresi, A.; et al. A Computer Vision Approach Based on Deep Learning for the Detection of Dairy Cows in Free Stall Barn. *Comput. Electron. Agric.* **2021**, *182*, 106030. [[CrossRef](#)]
28. Andrew, W.; Gao, J.; Mullan, S.; Campbell, N.; Dowsey, A.W.; Burghardt, T. Visual Identification of Individual Holstein-Friesian Cattle via Deep Metric Learning. *Comput. Electron. Agric.* **2021**, *185*, 106133. [[CrossRef](#)]
29. Borges Oliveira, D.A.; Ribeiro Pereira, L.G.; Bresolin, T.; Pontes Ferreira, R.E.; Reboucas Dorea, J.R. A Review of Deep Learning Algorithms for Computer Vision Systems in Livestock. *Livest. Sci.* **2021**, *253*, 104700. [[CrossRef](#)]
30. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot Multibox Detector. In Proceedings of the Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 11–14 October 2016; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer: Berlin/Heidelberg, Germany, 2016; Volume 9905 LNCS, pp. 21–37.
31. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; IEEE Computer Society: Washington, DC, USA, 2016; Volume 2016-Decem, pp. 779–788.
32. Redmon, J.; Farhadi, A. *YOLO9000: Better, Faster, Stronger*; Computer Vision Foundation: New York, NY, USA, 2017; pp. 7263–7271.
33. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. 2018. Available online: <https://arxiv.org/abs/1804.02767> (accessed on 18 April 2022).
34. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.

35. Gupta, H.; Varshney, H.; Sharma, T.K.; Pachauri, N.; Verma, O.P. Comparative Performance Analysis of Quantum Machine Learning with Deep Learning for Diabetes Prediction. *Complex Intell. Syst.* **2021**, *1*, 1–15. [[CrossRef](#)]
36. Kumar, S.; Yadav, D.; Gupta, H.; Verma, O.P.; Ansari, I.A.; Ahn, C.W. A Novel YOLOv3 Algorithm-Based Deep Learning Approach for Waste Segregation: Towards Smart Waste Management. *Electronics* **2021**, *10*, 14. [[CrossRef](#)]
37. Jindal, P.; Gupta, H.; Pachauri, N.; Sharma, V.; Verma, O.P. *Real-Time Wildfire Detection via Image-Based Deep Learning Algorithm*; Springer: Singapore, 2021; pp. 539–550.
38. Singh, L.K.; Pooja; Garg, H.; Khanna, M. Deep Learning System Applicability for Rapid Glaucoma Prediction from Fundus Images across Various Data Sets. *Evol. Syst.* **2022**, *1*, 1–30. [[CrossRef](#)]
39. Bhat, O.; Khan, D.A. Evaluation of Deep Learning Model for Human Activity Recognition. *Evol. Syst.* **2022**, *13*, 159–168. [[CrossRef](#)]
40. Vieira, S.M.; Kaymak, U.; Sousa, J.M.C. Cohen's Kappa Coefficient as a Performance Measure for Feature Selection. In Proceedings of the 2010 IEEE World Congress on Computational Intelligence (WCCI 2010), Barcelona, Spain, 18–23 July 2010. [[CrossRef](#)]
41. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposals. Available online: <https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf> (accessed on 18 April 2022).