Cow Teats Stall Number Analysis Using Object Detection

Preprint · November 2023

DOI: 10.13140/RG.2.2.26896.38400

1 author:

Deepshikha Mahato
Yeshiva University

Cow Teats Stall Number Analysis Using Object Detection

Deepshikha Mahato
Yeshiva University
dmahato@mail.yu.edu

Abstract

In this report, we present The CowStallNumbers dataset, containing 1042 training and 261 test images from cow teat videos, which accurately identifies and localizes cow stall numbers using a fine-tuned ResNet50 model and augmentation techniques like random crop, center crop, and random rotation. The dataset holds promise for advancing research in cow stall number detection and related domains, demonstrating the methodology's effectiveness.

1. Introduction

The dairy industry has experienced significant growth due to the increasing demand for milk and its derivatives. Maintaining cow health and productivity requires a comfortable environment, appropriate nutrition, and regular health assessments. A previous research focused on categorizing teats using phone images, but this method struggled to identify cows without recognizing stall numbers. A new model was developed to identify these stall numbers. The ResNet50 [4] model was fine-tuned for cow stall detection, and the method achieved exceptional accuracy in detecting cow stalls. This method enhances operational efficiency for dairy farmers and contributes to the sustainability and profitability of the dairy industry. The study introduced a small CowStallNumber dataset for stall number detection and fine-tuned the pre-trained ResNet50 [4] model, achieving 71% accuracy in stall number recognition and a 57.857%. Intersection over Union score in box position prediction.

2. Related Work

The Faster Region-based Convolutional Neural Network (R-CNN) [3] has shown promise in cow tail detection, delivering improved accuracy and faster detection rates. However, these methods face challenges in low lighting and low contrast environments, which can obscure cows' identifying features. Thermal imaging, using infrared technology, has emerged as a potential solution, capturing the heat signature of objects, and enabling more precise detection in low visibility or partially hidden environments. Detecting cows in dense herds remains challenging due to tight grouping and occlusion, and different cow breeds may require tailored detection algorithms.

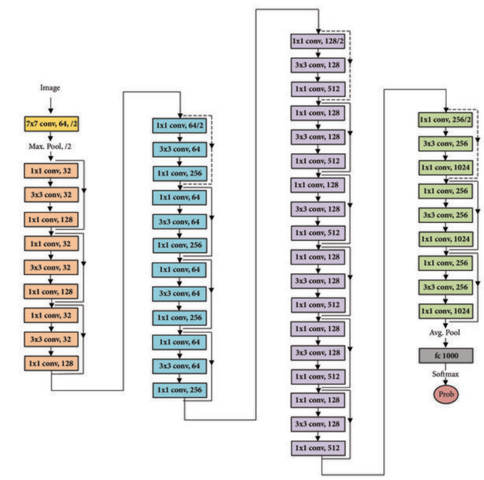

Figure 1. Model Architecture of ResNet-50

Refined cow detection methods have significant implications for farmers, enhancing herd management, ensuring appropriate care, nutrition, and disease control, and enhancing milk production. Progress in deep learning and thermal imaging technology is pivotal in accurately tracking cows and understanding their movements, but challenges persist. Other models include You Only Look Once (YOLO) [7] and Single Shot MultiBox Detector (SSD), RetinaNet, Mask R-CNN, Cascade R-CNN [2], Feature Pyramid Network (FPN) [6], DetNet [5], DenseBox, and others.

Evolutionary iterations within the YOLO (You Only Look Once) methodology have introduced notable advancements and modifications over time. YOLOv1 [7] surfaced in 2016 as the inaugural version [19]. Subsequent to its inception, YOLOv2 (also referred to as YOLO9000) emerged in 2017, leveraging the Darknet-19 network architecture alongside pivotal inclusions such as batch normal-

	imageFilename	box_position_1	box_position_2	box_position_3	box_position_4	class_names
0	GH030066_4679.png	287.0	116.0	25.0	23.0	29
1	GH020058_19383.png	275.0	58.0	23.0	24.0	44
2	GH030061_1269.png	306.0	74.0	26.0	25.0	13
3	GH020058_17235.png	NaN	NaN	NaN	NaN	0
4	GH020058_18319.png	288.0	55.0	23.0	23.0	42

Figure 2. Train.csv

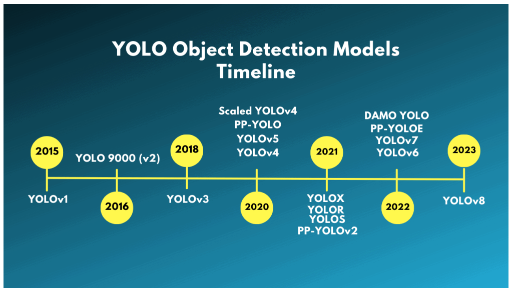

Figure 3. Evolution of YOLO Models

ization and anchor boxes. It introduced the ability to detect objects across diverse scales via multiscale predictions.

In 2018, the release of YOLOv3 marked a significant leap, employing the Darknet-53 architecture, integrating feature pyramid networks (FPNs), and leveraging residual connections. This iteration introduced dynamic scaling of anchor boxes, significantly enhancing object detection across varying scales.

The subsequent release, YOLOv4 in 2020, adopted the CSPDarknet-53 [1] architecture, incorporating novel features such as spatial pyramid pooling and an intricate data augmentation pipeline. YOLOv4 embraced efficient training approaches, including mosaic data augmentation and focal loss.

In contrast, YOLOv5, introduced as a PyTorch implementation, diverged from its origin, featuring a CSP backbone and PANET neck. Notable advancements included the implementation of auto-learning for bounding box anchors.

Lastly, the recent iteration, YOLOv6, unveiled in 2022, aimed at catering to industrial applications, boasting a streamlined single-stage object detection framework engineered for efficiency and high performance.

3. Model Building

A pre-trained ResNet-50 [4] model was implemented to identify cow stall numbers using the teats of the cows. A convolutional neural network design called ResNet-50 has demonstrated exceptional performance in a number of computer vision tasks. To provide a strong foundation for the extraction of features, the pre-trained model was initialized with weights derived from training on a large-scale dataset, usually ImageNet.

4. Fine-tuning and Training

Transfer learning [10] was used to modify the ResNet-50 model for the particular job of cow stall number detection. The last fully connected layer of the pre-trained model was added or changed to accommodate the number of classes that matched the different stall numbers found in the dataset. The adjusted model was then refined with an appropriate optimizer and loss function on the labeled cow teats dataset.

5. Experiment Setup

Hyperparameters, including learning rate, batch size, and the number of epochs, were fine-tuned through empirical experimentation to attain the best performance of the model.

6. Methods

The task involves object detection that includes multiple-class classification and accurate labeling. To enhance the model's effectiveness, a combined loss function was calculated for each batch. This composite loss function incorporates cross-entropy loss (Equation 1), smooth L1 loss (Equation 3), and IoU loss (Equation 5). The overall total loss (Equation 6) is utilized for refining the model's accuracy.

$$L_{ce} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{i,j}\log(p_{i,j}) \quad (1)$$

Equation 1 (denoted as L_{ce}) calculates cross-entropy loss, where N represents batch size, C indicates the number of classes, $y_{i,j}$ refers to the ground truth probability of the i^{th} sample belonging to class j, and $p_{i,j}$ signifies the predicted probability of the i^{th} sample belonging to class j.

$$l_i = \begin{cases} 0.5(\hat{yi} - y_i)^2, & \text{if } |\hat{yi} - y_i| < 1 \\ |\hat{yi} - y_i| - 0.5, & \text{otherwise} \end{cases} \quad (2)$$

Equation 3 (represented as l_i) computes smooth L1 loss, denoted as L_{SL1}, for individual samples in the batch.

$$IoU_i = \frac{|A_i \cap B_i|}{|A_i \cup B_i|} \quad (4)$$

Equation 4 (IoU_i) calculates the Intersection over Union (IoU) score between predicted and true values ($\hat{yi}$ and y_i) for the i^{th} element.

$$LIoU = \frac{1}{N}\sum_{i=1}^{N}(1 - IoU_i) \quad (5)$$

Equation 5 ($LIoU$) computes the mean IoU score across the entire batch.

$$L_{total} = L_{ce} + 0.1 \times L_{SL1} + 5 \times LIoU \quad (6)$$

Equation 6 (L_{total}) amalgamates cross-entropy, smooth L1, and IoU losses, adjusting their values with specific coefficients to align with L_{ce}. The optimization utilized the Adam optimizer with a weight decay of 5×10^4 and a learning rate.

7. Results

There were two main functions. First was initialize_resnet Function This function is responsible for setting up a ResNet-50 model for transfer learning. It performs the following tasks:

- **Loading Pre-trained ResNet-50 Model**: It utilizes the `models.resnet50(pretrained=True)` function to fetch the ResNet-50 model architecture pre-trained on the ImageNet [8] dataset. This pre-trained model comes with learned weights from the ImageNet dataset.
- **Freezing Base Layers**: If the parameter `freeze_base_layers` is set to `True`, it iterates through the parameters of the loaded ResNet-50 model and sets `requires_grad` to `False`. This prevents the base layers from being updated during training.
- **Modifying Output Layer**: Regardless of whether base layers are frozen or not, the final fully connected layer (also known as the classification layer) is replaced with a new layer (Linear layer) that matches the number of output classes specified (`num_classes`). This new output layer will adapt the model for the specific classification task it needs to perform.

And the second one was train_model Function. This function handles the training of the modified ResNet-50 model using the provided data loaders and settings. Here's a detailed breakdown:

- **Setting Device (GPU/CPU)**: The function checks if a device is specified. If not, it checks if a CUDA-enabled GPU is available; if available, it sets the device to GPU, otherwise to CPU.
- **Moving Model to Device**: The model is moved to the specified device (GPU or CPU) using the `model.to(device)` method. This ensures that all subsequent computations will be performed on the chosen device.
- **Training Loop**: It iterates through each epoch for a specified number of epochs (`num_epochs`). Within each epoch:

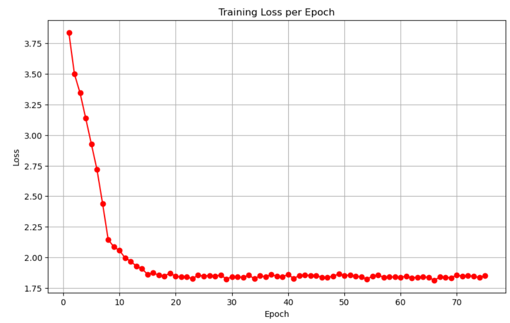

Figure 4. Plot of Test Accuracy v/s Loss Function

1. **Training Mode**: Sets the model to training mode using `model.train()`.
2. **Training Iterations**: Iterates through batches of data from the training set (`dataloaders['train']`). For each batch:
   - Moves the input data and labels to the specified device.
   - Performs a forward pass through the model to get predictions (`outputs`).
   - Calculates the loss between predicted outputs and actual labels using the specified criterion.
   - Computes gradients and updates model weights using the optimizer.
   - Keeps track of the running loss for the epoch.
3. **Learning Rate Adjustment**: Adjusts the learning rate using the scheduler based on the running average of the training loss.
4. **Recording Training Statistics**: Maintains a record of training loss for each epoch in the `stats` dictionary.
5. **Displaying Training Progress**: Prints the current epoch number and the corresponding training loss.

The code ran for Epoch 75/75, Loss: 1.8517092195424167, and gave accuracy on Test Data: 71.29% and the Intersection over Union (IoU) in percentage was: 57.857%.

8. Discussion

This section elucidates the comparative analysis of our study's outcomes employing the ResNet-50 architecture against the antecedent research conducted by [9]. Our research achieved an accuracy of 71.29% and an Intersection over Union (IoU) score of 57.857%, which indicates a proficient system in the classification of cow teat stall numbers. In contrast, [9] study, which utilized a similar approach, reported a slightly higher accuracy and IoU score. The differences in performance metrics can be attributed to various factors including dataset variability, model training nuances, and hyperparameter optimization.

While our model demonstrates a robust capability in identifying stall numbers accurately, the moderate IoU score points towards an area for improvement in the model's localization precision. [9] research, which yielded a higher IoU, suggests that their model was better at delineating the precise boundaries of the stall numbers. The discrepancy in localization accuracy could be due to differences in the training dataset's annotation quality, diversity in teat stall number presentation, or the employment of more sophisticated data augmentation techniques.

The current findings offer a pivotal extension to the body of knowledge established by [9], reinforcing the viability of deep learning models, like ResNet-50, in the agricultural domain. Nevertheless, our study underscores the need for ongoing advancements in model accuracy and localization efficacy to ensure comprehensive reliability in practical applications. Future studies should aim to bridge the gap in localization precision, perhaps through the integration of region proposal networks or advanced object detection frameworks that have shown promise in other domains.

In summary, both studies underscore the transformative potential of convolutional neural networks in agricultural automation, with the collective aim to enhance model performance for real-world deployment.

9. Conclusion

Our study employed the ResNet-50 architecture for the detection of cow teats stall numbers, achieving an accuracy of 71.29% and an IoU score of 57.857%. While our results exhibit promising accuracy rates, the IoU score suggests a moderate level of localization precision for identifying the stall numbers. This indicates that while the model is proficient in classification tasks, it may require further enhancements to precisely localize and delineate the stall numbers within the teats.

The attained accuracy signifies a reliable ability to identify the stall numbers, although the model's performance in accurately delineating the boundaries of these regions can be improved. Future research endeavors should focus on refining the model's localization capabilities, potentially through augmentation techniques, architectural modifications, or additional fine-tuning to further enhance the accuracy and robustness of the detection system.

Our findings provide a foundation for leveraging deep learning techniques, particularly the ResNet-50 architecture, for the detection of cow teats stall numbers. Despite the satisfactory accuracy achieved, the quest for higher precision and robustness in object localization remains a crucial area for improvement in this domain.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *Computer Science ¿ Computer Vision and Pattern Recognition*, Apr 2020. 2

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. 2017. 1

[3] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Computer Science ¿ Computer Vision and Pattern Recognition*, Nov 2013. 1

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Computer Science ¿ Computer Vision and Pattern Recognition*, 2015. Submitted on 10 Dec 2015. 1, 2

[5] Zeming Li, Chao Peng, Gang Yu, Xiangyu Zhang, Yangdong Deng, and Jian Sun. Detnet: A backbone network for object detection. 2018. 1

[6] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. 2017. 1

[7] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *Computer Science ¿ Computer Vision and Pattern Recognition*, Jun 2015. Last revised on 9 May 2016, version v5. 1

[8] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. 2014. 3

[9] Youshan Zhang. Stall number detection of cow teats key frames. *arXiv preprint arXiv:2303.10444*, 2023. 3, 4

[10] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. 2019. 2