Synthetically generated cow provides data for pose-estimation: computer graphics and deep learning for behavioral analysis

Ali Goldani[1,*], Navid Ghassemi[2,3], Karen Schwartzkopf-Genswein[4], Ian Q. Whishaw[5], Majid H. Mohajerani[3]

[1]The Centre for Neuroengineering Solutions, Lethbridge, Alberta, Canada.

[2]McGill University Integrated Program in Neuroscience, Montréal, Québec, Canada.

[3]Department of Psychiatry, Douglas Hospital Research Centre, McGill University, Montréal, Québec, Canada.

[4]Lethbridge Research and Development Centre, Agriculture and Agri-Food Canada, Lethbridge, Alberta, Canada.

[5]Canadian Centre for Behavioural Neuroscience, Department of Neuroscience, University of Lethbridge, Alberta, Canada.

Abstract

Bovine motor disorders can be difficult to diagnose and usually require human involvement. Artificial intelligence (AI) has introduced many tools in the field of computer vision (CV) that can help with this task, such as pose estimation models. Training such models requires substantial amounts of data. Given the harsh conditions of feedlots, gathering this data can prove to be challenging. To help with this issue, we propose a pipeline to generate this data synthetically. In this study, we used a three-dimensional digital representation of walking cattle to generate the data, compared its effectiveness with that of real footage captured from the real world, and trained pose estimation networks with them. By testing these networks on test data and analyzing their results, we concluded that this method could compensate for the scarcity of behavioral information and enable researchers to amend the shortcomings of their data. Furthermore, we illustrate another application for synthetic data, by using it to simulate a real-world problem such as finding the optimal placement of recording cameras in a feedlot and calculating the best parameters that can be involved.

Introduction

Motor disorders including lameness are prominent health concerns in feedlot cattle that affect their welfare, productivity, and profitability[1–3]. Lameness has a significant prevalence, accounting for about 16% of health problems in beef cattle, adding to production costs[4,5]. On average, feedlots lose approximately $60 USD per animal due to lameness[6], leading up to 70% of revenue loss[7]. Also, the treatment costs imposed to feedlot owners by each case of lameness range from $122 to $1391 CAD[2,8]. If lameness diagnosis can be made earlier and more precise, its duration could be decreased with a significant increase in cattle health and reduction in costs.

Lameness can be detected visually in animal movement. Hence, one of the primary methods for detecting lameness is using a scoring system to rate the degree of its severity. Feedlot personnel are trained to follow this procedure for scoring to detect lameness[1]. The problem with these rating methods is inconsistency in scoring since it requires the observer as the primary decision maker for scoring[9]. The degree of agreement between scorers will change over time based on their training and experience[10]. Transitioning from manual methods to more intelligent and automatic approaches can solve the inconsistency problem.

To detect lameness, the animal's gait pattern should be analyzed. To increase consistency in analyzing gait patterns, we need a method to measure gait patterns. There are multiple general approaches to do so. The first approach is using sensors. Since the 1980s, studies have investigated the potential of integrating sensors and analyzing their data to[11]. These studies have focused on measuring ground reaction force[12–14], automatic measurement of weight distribution[15,16], recording footsteps in a gait pattern[17], analyzing gait and/or activity using accelerometers[18,19]. Such approaches provide an accurate way of detecting lameness early on, but the sensor technologies used need to be very cost-efficient as some are hard to set up and take up considerable space.

A more modern approach is leveraging AI and CV to describe and analyze gait patterns. Several studies have already explored the potential of this approach. Applying classic methods of AI such as classification and regression algorithms to the integration of lameness scores and features of image recordings has shown to be effective in this task[20]. Currently, 34.1% of the studies focusing on lameness detection apply video analysis and image processing in their methods[21].

The advancements of AI and the ever-growing field of deep learning have revolutionized computer vision[22], creating tools such as object detection[23], and semantic segmentation[24]. One other tool that can significantly contribute to the matter at hand is pose estimation. Pose estimation provides information regarding the location of body parts given an image or a video of a subject. Through this method, a variety of problems concerning cattle behavior can be studied. There have been several promising works in this field which is constantly growing. Application of pose estimation extends from lameness detection[25,26] to activity classification[27] and weight estimation[28].

The first challenge in creating these models is that they require a substantial amount of data for training[29]. In comparison to studying rodents[30], acquiring data from a feedlot is more challenging. Animals are hard to control, and the environmental conditions cannot be completely controlled, since the subjects are roaming outdoors, in a variety of lighting conditions and weather. Moreover, even placing cameras in a stable place is not completely feasible due to extreme winds.

There are solutions to overcoming this limitation. Data augmentation[31], transfer learning[32], synthetic data generations using generative adversarial networks[33], and federated learning[34] are methods that solve problems using deep learning without requiring an extensive data collection approach. With new advancements in hardware design and their widespread accessibility, we can practice another form of synthetic data generation. By utilizing 3D graphical models in 3D modeling software or game engines, we can create realistic footage that resembles real-world data to an acceptable degree and can be used to provide ground truth data for training not only pose estimation models, but models for detection, and tracking[35,36].

In this project, we have used backgrounds from one-camera recordings of cattle walking in a single lane as a reference to create a 3D synthetic model of cattle behavior. Then we multiplied this data in numbers by altering the models to create more data and used them to train a supervised Deep Learning model. Our hypothesis is that the Deep Learning model trained with synthetic data in combination with real data will have a better performance in detecting the cattle pose than the model trained only with limited real data.

Results

To evaluate the effectiveness of using synthetic data generated from three-dimensional (3D) models in training neural networks for cattle pose estimation, we designed a scenario in which we generate a synthetic dataset to complement a dataset recorded from real-world settings. For evaluation, we tested the performance of models trained on these datasets separately and combined. In this section, we present the evaluation metrics and elaborate on them.

Table 1 shows COCO average precision and COCO average recall calculated for each of these models. We have performed a 5-fold cross-validation training with 20% of data being withheld from the models in each training. The models are then tested on a test dataset that has not been present in its training and validation. These metrics are averaged across all the models' training.

Real and synthetic models are close to each other in terms of average precision, with the synthetic model having slightly higher precision. This shows that the synthetic data is designed properly and encapsulates the features of real-world scenarios, which confirms that our experiment succeeded in targeting the required variety of data. Based on this, we can conclude that synthetic data can even replace real data in such scenarios. The combined model improves the precision of the real model while also increasing its recall. This means that by seeing the synthetically generated data after learning the foundations of real data, the model was able to generalize more in its predictions and make better detections.

To further analyze these models, we can take a look at the performance of each keypoint. Figure 6a includes polar plots that show the average object keypoint similarity score of each keypoint for all models. The model with a higher coverage of the surface of the polar plot has a higher performance in predicting keypoints. In upper keypoints including points on the back, face, and neck, we can see that the models perform very closely. The combined model has taken the best features of each model and covers the largest area. In lower keypoints, it is evident that detecting the R_F_Paw keypoint (Right front paw) has been challenging, especially for the real model. In this case, the combined model has tried to learn from synthetic data while keeping the bias with the real model.

By taking a look at Figure 6b, the difference between models becomes more apparent. On average, the combined model has the least pixel distance between its predictions and ground truth. Also, in Figure 6c, we can see that the combined model has fewer outliers than the real model and is more consistent in its predictions than the synthetic model. Figure 6f, Figure 6g, and Figure 6h show the attention of each model in the form of a heat map, we can see that the synthetic model is complimentary to the real model, (performing not as well in back arcs and better in paws). The combined model takes features from both models to reach a satisfactory performance. This makes the combined model a good choice for our work.

These results suggest that the process of generating synthetic data for training neural networks can be effective for the analysis of cattle movements. By using 3D models to generate synthetic data, we can quickly and cost-effectively generate large amounts of training data, which can improve the accuracy and efficiency of behavior analysis. However, synthetically generated data cannot entirely replace the use of real-world data in this matter. As shown before, the model reaches its best performance when it has seen real data to form its foundation and keeps a connection with real-world scenarios.

Discussion

In an era characterized by automation and AI, industries that fail to leverage these technologies risk falling behind. The cattle industry is no exception. One of the primary applications of AI in this sector is the diagnosis of motor disorders, the importance of which we have discussed in this paper. However, challenges associated with data collection remain significant obstacles, which this work seeks to address. Data collection is a resource-intensive and expensive process, and insufficient data prevents effective model training.

Our work proposes an alternative solution to reduce data collection costs by generating a synthetic feedlot environment. We demonstrated that synthetic data enables the integration of variations—such as changes in weather conditions, backgrounds, and subject characteristics—in datasets much faster than natural data collection. Collecting natural data across different weather conditions can require months of waiting for the appropriate season, assuming data collection is even feasible during that time. We further showed that these synthetic datasets are of high quality, enabling few-shot learning[37] with synthetic data, and can be utilized for the initial training of models. Moreover, we demonstrated that synthetic data can improve model performance and, more importantly, enable model generalization by simulating varying conditions and accounting for different environmental changes.

We selected pose estimation as the platform for our initial experiments, as it is widely accepted in the behavioral studies literature as a method for automatically analyzing movement and behaviors[38]. The results of this research have the potential to advance the wider field of cattle behavioral studies. By overcoming limitations posed by scarce training data, deep learning models can be developed and utilized to analyze specific behavioral patterns. For example, pose estimation models can be combined with statistical analyses to enable early diagnosis of lameness and other motor disorders in feedlot cattle, which could significantly improve animal welfare, increase productivity, and reduce economic losses.

In addition to assisting with model training, these synthetic environments can significantly contribute to every step of solution development and simulation. When designing a product—specifically for monitoring applications—there are numerous variables to consider, such as the positioning, number, and quality of cameras. Subsequent rounds of stress testing are necessary to ensure that the collected data meet the quality requirements for the intended task.

Moreover, there are many tasks to consider in feedlot monitoring with cameras beyond motor disorder diagnosis. Tasks such as weight estimation[28] (to monitor growth), activity monitoring[27,39–42], feeding behavior analysis[43], and calving event prediction[44] have all shown promise in automation but require extensive testing. With synthetic data, users can first determine the optimal settings for placing their data collection devices and, second, perform multiple rounds of stress testing synthetically. This approach is much more cost-effective than conducting tests in a natural environment and allows for the identification and correction of issues before deployment.

In Figure 7, you can see an example of such a simulation. By recording a single action of a cow from multiple cameras in the 3D environment, and training models or performing analysis based on them, we can find suitable recording conditions in the real world and be confident that we will acquire satisfactory results.

In an example of the weight estimation research, it is expected for the model to have the highest efficiency given the top camera footage, as it aligns with data used in the literature[28].

Despite the promising results, there are still some limitations in the methodology of our work, which opens the door for future studies to address them. Firstly, we confined our testing to pose estimation due to its widespread acceptance in the literature. Future research should explore the quality of synthetic data—and the potential to replace or reduce the need for real data—in other applications of AI in the cattle industry, such as body segmentation[45] or tracking[46].

Secondly, and more importantly, although we have aimed to eliminate many overheads of data collection and related expenses, generating synthetic data still requires extensive knowledge of various computer graphics software and a solid understanding of cattle behavior and interactions. To address this, we have transformed our models into a user-friendly toolbox and made them publicly available on GitHub. This resource is intended to assist individuals with minimal knowledge of computer graphics in utilizing our methods. For the second aspect, we plan to develop an environment with multiple subjects that can interact, incorporating the dynamics of their behaviors into our model in future work.

To summarize, in this work, we recognized pose estimation models as potential AI diagnostic tools. We then showed that utilizing 3D modeling and synthetic data generation can help in creating such models by reducing the time and resources required to gather their training data. This is an exciting precedent for the potential of synthetic data generation in the field of AI and Deep Learning applications, particularly in areas where the collection of large, diverse, and accurately labeled real-world data is a challenge. Given the efficiency of synthetic data, we propose applying them in simulations preceding real-world data capture.

In these simulations, many environmental parameters can be adjusted and analyzed, allowing the researchers to choose the best ones without significant amounts of time and effort.

Methods

Data Acquisition

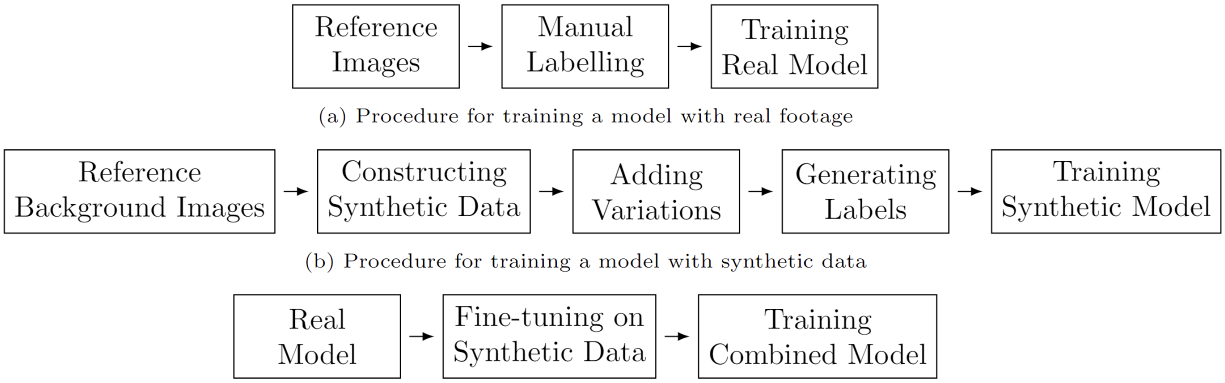
The real footage from feedlot cattle used in this project is acquired through The Lethbridge Research and Development Centre led by Karen Schwartzkopf-Genswein. The videos are captured using a GoPro Hero 10. The camera is set up in front of a walking aisle for cattle and cattle are led to pass through that aisle while the camera is recording. It is positioned in a constant position during the recording with a varying distance of 1 to 3 meters from the cattle. Weather conditions in the recordings vary between sunny, cloudy, or snowy, changing the light effects and shadows in between the recordings. The subject cattle appear in colors black and brown and with patterns of white spotting.

Preprocessing

The camera used to record the footage from the feedlot uses an ultra-wide lens. This kind of lens introduces a distortion to the picture, especially around the edges of the image. To make this footage represent reality as best as possible, a simple distortion-fix algorithm is applied to the footage before using it for the next steps. The code for this algorithm is provided in the supplementary information.

Pipelines for Creating Deep Learning Models

To illustrate the effects of using a 3D cow model based on our hypothesis, we train three different neural networks using different pipelines shown in Figure *1*. These pipelines are described in the following. The real model is created using real footage acquired from the feedlot with manual labeling and no modifications applied to the data. This is the classical pipeline for training a DL model (Figure 1a). The synthetic model pipeline is similar to the real model pipeline but with synthetic data and generated labels instead of real data and manual labeling. The synthetic data contains subjects similar to real data, but with various backgrounds from real environments and a combination of changes applied to their appearance so that it would have more variety. In comparison, we expect this method not to be as efficient as the real model as it will be evaluated on real data only (Figure 1b). The combined model is the pipeline for augmenting data to fill the gaps in real footage. Here, the real model undergoes a fine-tuning process using the data for training the synthetic model. This way, the model that has formed an understanding of what real data is, learns a variety of features in the appearance of subjects. We hypothesize that this pipeline will have an increased performance in evaluation. (Figure 1c)



(a) Procedure for training a model with real footage

(b) Procedure for training a model with synthetic data

(c) Procedure for training a model with real footage and fine-tuning it with synthetic data

Figure 1. Pipelines used to train models. Procedures for training a model with a) real footage b) synthetic data c) real footage and fine-tuning it with synthetic data.

Creation of Cow Model

For the base model of the cow, we use a proprietary artist-created model that contains default animations for generic behaviors such as walking[47]. It is a generic cow model that resembles the appearance of the typical feedlot cattle. Figure *2* depicts the elements of this cow model. Figure *2*b shows the basic appearance of this model. It is created based on the anatomical features of an average feed cattle. This model can be extended with hair and skin simulation and provides the ability to implement physics. By using the real cow in the footage as a reference, Figure *2*a, we match the size, and shape and use texture modification to change the color of the cow's coating (including spotting) to create patterns close to what we expect to see in real life. To be able to change the model's appearance and move its components in the way that a real cow does, we need to have the bone structure for the cow. This process is called Rigging. Figure ***Figure 2***d shows the armature included in this model to control the model's soft-body component. Using an atlas for cow anatomy, Figure *2*c, we have matched the bone structure to make sure it is an exact representation of a cow's anatomy. Finally, hair particles and textures make the realistic synthetic cow shown in figure Figure *2*e. This version is used in the data augmentation process.

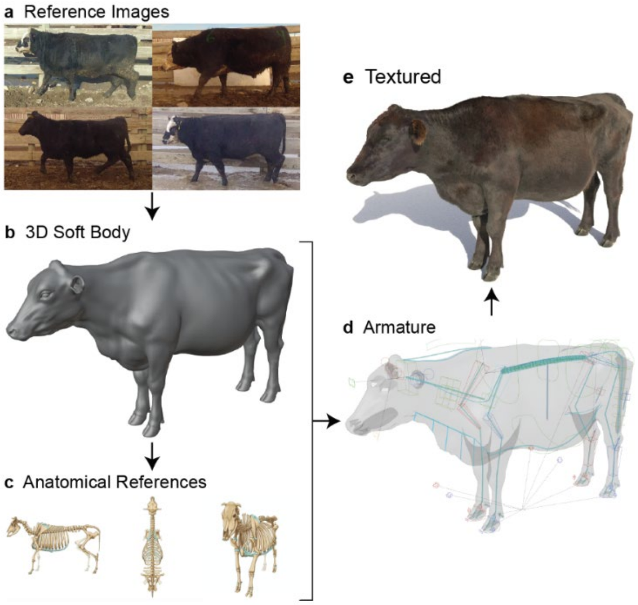


Figure 2. Process for creating the base 3D model of the cow. a) Reference pictures for creating manual animation. These pictures are frames taken from available video footage. b) The base 3D model of the cow that we use for feature-matching with the cows in our footage. c) Anatomical Reference of the cow (reference). d) Added armature to the cow based on the anatomical reference. This armature plays the role of the skeleton of the animal and is used in the animation process. e) Textured version of the cow with all the added details to reflect the visual appearance of a real cow.

Scene Matching and Animation

The model is now imported into a scene in Blender. We take a series of background images belonging to real recording environments, then all environment details such as lighting (angle and the intensity of the light), environment elements (background objects and flooring), and camera specifications (distance from subject, field of view, and focal length) are adjusted in the scene to match that background. For example, the color and angle of the environment lighting are modified in a way to illuminate the subjects the same as the background and cast shadows with length, angle, and intensity similar to the background. Following this method, we create a synthetic version of the real video that matches the real footage.

The acquired 3D model comes with a set of pre-made animations that display cycles of movements of a cow (walking, trotting, eating, etc.). By adding changes to the position and the path of the animation, we create varied movements that help us build the augmented dataset.

Synthetic Data Augmentation

To create more data that can be used for Deep Learning model training, we make changes to the synthetic animation we created in the previous step. These changes are applied to:

- Cattle appearance: In Blender, the color and texture of objects can be controlled through a tool-set called geometry nodes. Using geometry nodes, we can shift the color of the cow naturally by adding shades of color and merging them in natural-looking patterns. Also, using these nodes, we can generate a randomized spotting pattern on the cow to create an even more realistic coating.
- Lighting: Lighting details can be changed in the blender scene to change the time of day for the synthetic footage, having different lighting conditions in the videos can provide more variety in the training dataset for Deep Learning models.
- Viewing angle: As the synthetic model is a three-dimensional model, we can change the viewing angle to create synthetic data that would be equivalent to recording the real video by putting the camera in a different position.
- Environment: Using Blender, we can create elements that exist in the background in the 3D space so that they cast their shadows and display the shadow of the cow that is cast on them. This way, we can even recreate complex scenarios in which the shadow cast by the cow can be mistaken for its limbs due to recording conditions.

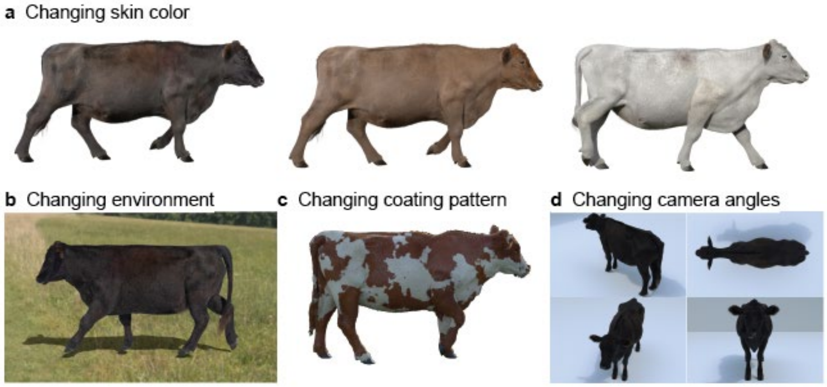


Figure 3. Steps in Adding Variations to the augmented cow. a) The basic model comprised of only the soft body is given a new natural skin color. b) The cow model in a) in a different augmented environment. c) The cow model in a) with a randomly generated coating. d) Display of different angles available for data augmentation.

By using combinations of the above-mentioned changes, we can create several videos only based on a single scene setting which can significantly help with the lack of data in training Deep Learning networks.

Realistic Rendering

After defining the scenes that would match the backgrounds of real recordings and setting up the surrounding objects to interact with the environment, we can modify elements in them to introduce randomized variations to our augmented dataset. To do this we use a set of parameters that can be interpreted in Blender as scene settings. After all the parameters have been set, the Python script that we have developed, goes through them, making changes and preparing for render. We use the high-fidelity rendering capabilities of Blender to generate life-like images that resemble a real cow in the created condition. We use Cycles Render Engine in Blender to render the footage. The generation of each frame takes between 1 to 3 minutes (frames that require a repeat of physical simulation take longer) using our hardware at the time of writing this paper (AMD Ryzen5 5600x CPU, Nvidia RTX 4080 16GB GPU).

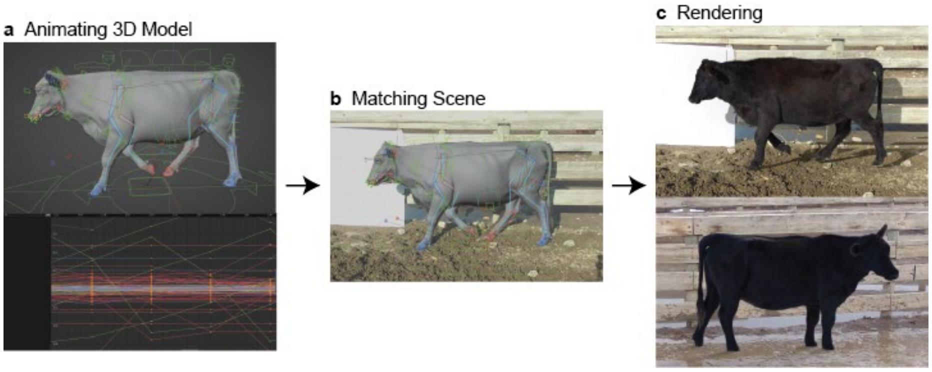


Figure 4. Steps in Constructing Synthetic Data. a) Process of animating the soft body based on the real-world footage, rigging view at the top and animation graph at the bottom panel. b) Using the real-world footage background with the augmented data. c) Result of realistic rendering process of cow models that include realistic coating, fur, and other features.

Automatic Labeling and Data Export

In this work, we are using a 20-keypoint model (cow-20kp) inspired by the keypoint schemas used in Ap-10k[48] and AnimalPose[49] datasets with more focus on the back arc of the cow (Figure 5). For the labels of our datasets, we have chosen MSCOCO format introduced by Microsoft Research[50] because of its popularity among the scientific works focusing on object detection, segmentation, and keypoint detection (our use case). This annotation format includes information about keypoints such as position, visibility and place in object structure.

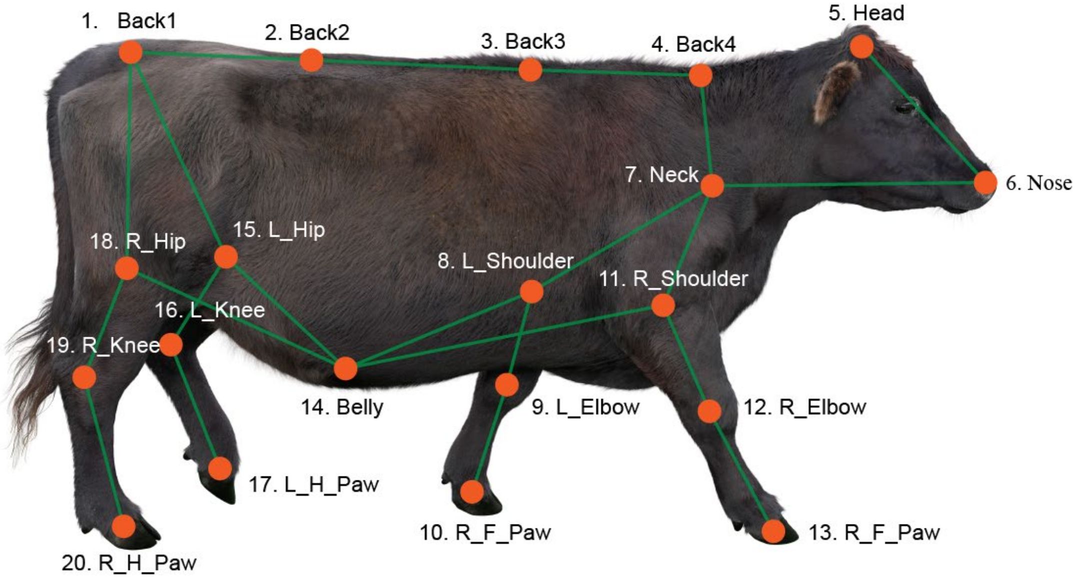


Figure 5. Display of 20 keypoints used in this study to describe the pose of a cattle. 1-4: Back arc points, 5: Head, 6: Nose or Snout, 7: Neck, 8: Left Shoulder, 9: Left Elbow, 10: Left Front Paw, 11: Right Shoulder, 12: Right Elbow, 13: Right Front Paw, 14: Belly, 15: Left Hip, 16: Left Knee, 17: Left Hind Paw, 18: Right Hip, 19: Right Knee, 20: Right Hind Paw. In the synthetic cow model, there is a marker object associated with each of these keypoints which is used for the automatic extraction of labels for each frame.

In training the real model, data annotation is performed using CVAT[51] which is a time-consuming task. On the contrary, the labels and required information from the synthetic models are readily available due to the possibility of tracking objects in Blender. However, since all the labels are present in our model, we need to reduce them to the ones only visible from the point of view of the camera. To do so, we use a simple technique called Ray Casting in Computer Graphics. In this method, we check if a direct line of sight exists between the camera in the environment and the marker that we want to track. As a result, points that are occluded or out of view can be detected and properly handled.

In ray casting, given two points $P_1 = (x_1, y_1, z_1)$, 3D position of the camera in the environment, and $\mathrm{P}_2 = (\mathrm{x}_2, \mathrm{y}_2, \mathrm{z}_2)$, 3D position of the marker on the cow's body, the equation of the ray $R(t)$ that starts at P_1 and passes through P_2 is given by:

$$R(t) = P_1 + t(P_2 - P_1) \tag{1}$$

Where:

- $R(t)$ is a point on the ray.
- t is a parameter. When $t =, R(t) = P_1$, and when $t =, R(t) = P_2$. For values of t between 0 and 1, $R(t)$ will be a point on the line segment between P_1 and P_2. For $t < 0$ or $t > 1$, $R(t)$ will be on the ray but outside the line segment.
- P_1 and P_2 are vectors representing the coordinates of the points.

We check for values of t between 0 and 1 to see if there is a point that fits in the equation. If the calculated value for t is anything other than 0 or 1, it means that there are points breaking the line of sight from the camera to the point. Thus, we omit the marker at point P_2 from the exported list of labels. After the rendering process is completed, we use custom scripts (see supplementary information) to export information such as joint positions in 2D.

Experiment Design

The main goal of this study is to show that with the current capabilities of 3D graphics tools, we can minimize the need for in-field data gathering. To show this, we use limited footage of real data, lacking variety in subject skin color, lighting condition, and environment to train a model for the task of pose estimation on cattle. Then we test this model's performance on a subset of data that is very different from its own. Then using our synthetic data generation method described in this work, we create a subset of data that will have a higher range of variety and use it to train a synthetic model. By testing these models and a third model resulting from their combination, we can determine the effectiveness of synthetic data.

Model Training

We chose to use the MMpose framework[52] as an approved standard in neuroscience communities due to its ease of use and the number of current pre-trained models in its model zoo.

Following a good practice in solving Artificial Intelligence problems, we do not perform the training from the ground up when possible. We use a pre-trained model on the Ap-10k dataset using the HRNet backbone and fine-tune it on our data with the cow-20kp keypoints schema. Approaching industry-grade performance in a short time is more possible this way. Now, for training real and synthetic models, we continue the training process from the last checkpoint--a saved state during the previous training--of the pre-trained network, so that the model learns to map its representation of cow pose, which it has acquired over the process of training on Ap-10k dataset, onto a 20-output head (one output for each keypoint). As for the combined model, we want to see if the gaps that exist in real data can be filled with synthetic data. To do so, we take the model trained on real data and fine-tune it by resuming its training on synthetic data. Table Table *2* describes the data used for training each model.

Evaluation

To ascertain the robustness and generalization capabilities of the trained models, it is pivotal to test their performance on data they have never encountered before. This not only provides a benchmark for their reliability but also offers insights into how effectively the augmented data mimics real-life scenarios, especially when compared with models trained exclusively on real data.

For the evaluation phase, a set of unique frames from cattle walking has been chosen that differs from the training dataset of each model and holds the variety that can be observed in a feedlot. This ensures that a good performance result can mean a good performance in real-world scenarios. This dataset is then labeled and verified to create the ground truth for calculating evaluation metrics.

In the evaluation process, the test dataset is given to all models, and the pose that they have predicted for each frame is compared to the ground truth labels to generate the evaluation metrics. The evaluation metrics are described below.

Precision and Recall

Precision and recall are two of the most commonly used metrics for evaluating performance in detection tasks. Precision is defined as the proportion of correct detections out of all the detections that the model has made. A high precision value indicates that the model makes accurate predictions with few mistakes. Recall, on the other hand, measures the proportion of correct detections out of all the ground truth instances. A model with high recall successfully detects most of the relevant instances, resulting in fewer missed detections. Relying on Precision alone can lead to favoring a model that only makes a few correct detections and misses the rest. Paying attention solely to Recall can also result in choosing a model that makes many incorrect detections as well as correct detections. Hence, it is crucial to consider both of these values at the same time; a model with high precision and recall metrics is ideal.

Following MS COCO guidelines for keypoint estimation tasks, we use MS COCO Average Precision (AP) and Recall (AR) as the metrics to report the performance of our models. In this method of calculating AP and AR, object keypoint similarity (OKS) is defined to determine the validity of a keypoint prediction.

$$OKS = \frac{\sum_i exp\left(\frac{-d_i^2}{2s^2k_i^2}\right)\delta(v_i > 0)}{\sum_i \delta(v_i > 0)}, \quad 0 \leq OKS \leq 1, \quad v_i \in \{0,1,2\} \tag{2}$$

Where:

- d_i is the Euclidean distance between prediction and ground truth value of keypoint i.
- s denotes object's scale, it is considered as the square root of the object's area.
- k_i is a constant specific to keypoint i that controls falloff, which determines how sensitive OKS is to d_i. Larger k_i increases the tolerance of OKS, meaning that the prediction can have more distance from the ground truth before causing a drop in OKS.
- v_i specifies if keypoint i is visible. $v_i = 0$ means that keypoint i is not in the image. $v_i = 1$ means that keypoint i is in the image, but it is somehow occluded. $v_i = 2$ shows that the keypoint is completely visible. Only keypoints with $v_i > 0$ impact OKS.

Given the OKS, COCO AP and COCO AR can be calculated using equations (3) and (4). COCO AP and COCO AR are calculated across a range of OKS thresholds, from 0.5 to 0.95 in steps of 0.05.

$$COCO/AP = \frac{1}{|\text{OKS Thresholds}|} \sum_{t \in \text{OKS Thresholds}} AP_t \tag{3}$$

$$COCO/AR = \frac{1}{|\text{OKS Thresholds}|} \sum_{t \in \text{OKS Thresholds}} AR_t \tag{4}$$

Average Pixel Distance and Average Relative Error Percentage

To illustrate the performance of these models in a more intuitive way, we use two metrics to show the pixel difference between prediction and ground truth. Pixel distance shows exactly how far the prediction is from ground truth on a given image. Since our test data is consistent in pixel dimensions, this metric can be used as a measure for comparison. For calculating the error percentage, we need to specify a base for normalizing all the distances. It is a common practice to use the head length of the animal (distance between Head and Nose keypoints in this case). Equations (5) and (6) result in these metrics.

$$AveragePixelDistance = \frac{\sum_i d_i \delta(v_i > 0)}{\sum_i \delta(v_i > 0)} \tag{5}$$

$$AverageErrorPercentage = \frac{\sum_i \frac{d_i}{d_{H_i}}}{\sum_i \delta(v_i > 0)} \tag{6}$$

Where:

- d_{H_i} is the length of the head in the image.
- Other parameters are the same as in equation (2).

Data availability

TODO

Code availability

All of the developed code, tools, and software components are available at:

https://github.com/Mohajerani-Lab/mmpose-synthetic-tune

References

1. Terrell, S. P., Reinhardt, C. D., Larson, C. K., Vahl, C. I. & Thomson, D. U. Incidence of Lameness and Association of Cause and Severity of Lameness on the Outcome for Cattle on Six Commercial Beef Feedlots. *Journal of the American Veterinary Medical Association* **250**, 437–445 (2017).
2. Davis-Unger, J. *et al.* Economic Impacts of Lameness in Feedlot Cattle1. *Translational Animal Science* **1**, 467–479 (2017).
3. Fitzsimmonds, H. M. Survey Assessing Foot Trimmer Involvement in Managing Lameness in UK Beef Cattle. *Veterinary Record* **195**, (2024).
4. Terrell, S. P. *et al.* Perception of Lameness Management, Education, and Animal Welfare Implications in the Feedlot From Consulting Nutritionists, Veterinarians, and Feedlot Managers. 2013 (2013) doi:10.21423/aabppro20134224.
5. Lhermie, G. *et al.* Economic Effects of Policy Options Restricting Antimicrobial Use for High Risk Cattle Placed in U.S. Feedlots. *Plos One* **15**, e0239135 (2020).
6. Erickson, S. E., Booker, C. W., Jelinski, M. & Janzen, E. D. The Epidemiology of Hoof-Related Lameness in Western Canadian Feedlot Cattle. *American Association of Bovine Practitioners Conference Proceedings* 11–14 (2023) doi:10.21423/aabppro20228587.
7. Cortés, J. A., Hendrick, S., Janzen, E. D., Pajor, E. A. & Orsel, K. Economic Impact of Digital Dermatitis, Foot Rot, and Bovine Respiratory Disease in Feedlot Cattle. *Translational Animal Science* **5**, (2021).
8. Wong, N. S. T. Characterization of the Hoof Bacterial Communities in Feedlot Cattle Affected With Digital Dermatitis, Foot Rot or Both Using a Surface Swab Technique. *Animal Microbiome* **6**, (2024).

9. Burgstaller, J., Wittek, T., Sudhaus-Jörn, N. & Conrady, B. Associations between Animal Welfare Indicators and Animal-Related Factors of Slaughter Cattle in Austria. *Animals* **12**, 659 (2022).

10. Tunstall, J., Mueller, K., Grove White, D., Oultram, J. W. H. & Higgins, H. M. Lameness in Beef Cattle: UK Farmers' Perceptions, Knowledge, Barriers, and Approaches to Treatment and Control. *Front. Vet. Sci.* **6**, 94 (2019).

11. Hogeveen, H., Kamphuis, C., Steeneveld, W. & Mollenhorst, H. Sensors and clinical mastitis--the quest for the perfect alert. *Sensors (Basel)* **10**, 7991–8009 (2010).

12. Rajkondawar, P. G. *et al.* The development of an objective lameness scoring system for dairy herds: pilot study. *Transactions of the ASAE* **45**, 1123 (2002).

13. Rajkondawar, P. G. *et al.* A system for identifying lameness in dairy cattle. *Applied engineering in agriculture* **18**, 87 (2002).

14. Rajkondawar, P. G. *et al.* Comparison of models to identify lame cows based on gait and lesion scores, and limb movement variables. *J Dairy Sci* **89**, 4267–75 (2006).

15. Neveux, S., Weary, D. M., Rushen, J., Von Keyserlingk, M. A. G. & De Passillé, A. M. Hoof discomfort changes how dairy cattle distribute their body weight. *Journal of Dairy Science* **89**, 2503–2509 (2006).

16. Pastell, M. *et al.* Assessing cows' welfare: Weighing the cow in a milking robot. *Biosystems engineering* **93**, 81–87 (2006).

17. Maertens, W. *et al.* Development of a real time cow gait tracking and analysing tool to assess lameness using a pressure sensitive walkway: The GAITWISE system. *Biosystems engineering* **110**, 29–39 (2011).

18. Chapinal, N. *et al.* Measurement of acceleration while walking as an automated method for gait assessment in dairy cattle. *J Dairy Sci* **94**, 2895–901 (2011).

19. Pastell, M., Tiusanen, J., Hakojärvi, M. & Hänninen, L. A wireless accelerometer system with wavelet analysis for assessing lameness in cattle. *Biosystems engineering* **104**, 545–551 (2009).

20. Coşkun, G., Şahin, Ö., Delialioğlu, R. A., Altay, Y. & Aytekin, İ. Diagnosis of lameness via data mining algorithm by using thermal camera and image processing method in Brown Swiss cows. *Trop Anim Health Prod* **55**, 50 (2023).

21. Nejati, A., Bradtmueller, A., Shepley, E. & Vasseur, E. Technology applications in bovine gait analysis: A scoping review. *PLoS One* **18**, e0266287 (2023).

22. Voulodimos, A., Doulamis, N., Doulamis, A. & Protopapadakis, E. Deep Learning for Computer Vision: A Brief Review. *Computational Intelligence and Neuroscience* **2018**, 1–13 (2018).

23. Zhao, Z.-Q., Zheng, P., Xu, S.-T. & Wu, X. Object Detection With Deep Learning: A Review. *IEEE Trans. Neural Netw. Learning Syst.* **30**, 3212–3232 (2019).

24. Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V. & Garcia-Rodriguez, J. A Review on Deep Learning Techniques Applied to Semantic Segmentation. Preprint at http://arxiv.org/abs/1704.06857 (2017).

25. Barney, S., Dlay, S., Crowe, A., Kyriazakis, I. & Leach, M. Deep learning pose estimation for multi-cattle lameness detection. *Sci Rep* **13**, 4499 (2023).

26. Li, Z. *et al.* Fusion of RGB, optical flow and skeleton features for the detection of lameness in dairy cows. *Biosystems Engineering* **218**, 62–77 (2022).

27. Wei, Y. *et al.* Study of Pose Estimation Based on Spatio-Temporal Characteristics of Cow Skeleton. *Agriculture* **13**, 1535 (2023).

28. Liu, H., Reibman, A. R. & Boerman, J. P. Feature extraction using multi-view video analytics for dairy cattle body weight estimation. *Smart Agricultural Technology* **6**, 100359 (2023).

29. Emam, Z. *et al.* On The State of Data In Computer Vision: Human Annotations Remain Indispensable for Developing Deep Learning Models. Preprint at http://arxiv.org/abs/2108.00114 (2021).

30. Kahnau, P. *et al.* A systematic review of the development and application of home cage monitoring in laboratory mice and rats. *BMC Biol* **21**, 256 (2023).

31. Yang, S. *et al.* Image Data Augmentation for Deep Learning: A Survey. Preprint at http://arxiv.org/abs/2204.08610 (2023).

32. Zhuang, F. *et al.* A Comprehensive Survey on Transfer Learning. *Proc. IEEE* **109**, 43–76 (2021).

33. Goodfellow, I. *et al.* Generative adversarial networks. *Commun. ACM* **63**, 139–144 (2020).

34. Zhang, C. *et al.* A survey on federated learning. *Knowledge-Based Systems* **216**, 106775 (2021).

35. Bolaños, L. A. *et al.* A three-dimensional virtual mouse generates synthetic training data for behavioral analysis. *Nat Methods* **18**, 378–381 (2021).

36. Plum, F., Bulla, R., Beck, H. K., Imirzian, N. & Labonte, D. replicAnt: a pipeline for generating annotated images of animals in complex environments using Unreal Engine. *Nat Commun* **14**, 7195 (2023).

37. Parnami, A. & Lee, M. Learning from Few Examples: A Summary of Approaches to Few-Shot Learning. Preprint at http://arxiv.org/abs/2203.04291 (2022).

38. Mathis, M. W. & Mathis, A. Deep learning tools for the measurement of animal behavior in neuroscience. *Current Opinion in Neurobiology* **60**, 1–11 (2020).

39. Gong, C. *et al.* Multicow pose estimation based on keypoint extraction. *PLoS ONE* **17**, e0269259 (2022).

40. Li, Z., Song, L., Duan, Y., Wang, Y. & Song, H. Basic motion behaviour recognition of dairy cows based on skeleton and hybrid convolution algorithms. *Computers and Electronics in Agriculture* **196**, 106889 (2022).

41. Khin, M. P., Zin, T. T., Mar, C. C., Tin, P. & Horii, Y. Cattle Pose Classification System Using DeepLabCut and SVM Model. in *2022 IEEE 11th Global Conference on Consumer Electronics (GCCE)* 494–495 (IEEE, Osaka, Japan, 2022). doi:10.1109/GCCE56475.2022.10014248.

42. Yang, Y., Komatsu, M., Oyama, K. & Ohkawa, T. SCIRNet: Skeleton-based cattle interaction recognition network with inter-body graph and semantic priority. in *2023 International Joint Conference on Neural Networks (IJCNN)* 1–8 (IEEE, Gold Coast, Australia, 2023). doi:10.1109/IJCNN54540.2023.10191592.

43. Islam, M. N., Yoder, J., Nasiri, A., Burns, R. T. & Gan, H. Analysis of the Drinking Behavior of Beef Cattle Using Computer Vision. *Animals* **13**, 2984 (2023).

44. Aoki, M. & Sugiura, R. Image-based Estimation of Pelvic Skeletal Key Points of Dairy Cattle by Using Deep Learning. *Agricultural Information Research* **33**, 59–64 (2024).

45. Jiang, B. *et al.* FLYOLOv3 deep learning for key parts of dairy cow body detection. *Computers and Electronics in Agriculture* **166**, 104982 (2019).

46. Gardenier, J., Underwood, J. & Clark, C. Object Detection for Cattle Gait Tracking. in *2018 IEEE International Conference on Robotics and Automation (ICRA)* 2206–2213 (IEEE, Brisbane, QLD, 2018). doi:10.1109/ICRA.2018.8460523.

47. Black Cattle Animated 3D - TurboSquid 1907491. https://www.turbosquid.com/3d-models/black-cattle-animated-3d-1907491.

48. Yu, H. *et al.* AP-10K: A Benchmark for Animal Pose Estimation in the Wild.

49. Cao, J. *et al.* Cross-Domain Adaptation for Animal Pose Estimation. in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* 9497–9506 (IEEE, Seoul, Korea (South), 2019). doi:10.1109/ICCV.2019.00959.

50. Lin, T.-Y. *et al.* Microsoft COCO: Common Objects in Context. in *Computer Vision – ECCV 2014* (eds. Fleet, D., Pajdla, T., Schiele, B. & Tuytelaars, T.) vol. 8693 740–755 (Springer International Publishing, Cham, 2014).

51. CVAT.ai Corporation. Computer Vision Annotation Tool (CVAT). (2024) doi:10.5281/zenodo.12771595.

52. MMPose Contributors. OpenMMLab Pose Estimation Toolbox and Benchmark. (2020).

Acknowledgments

Author contributions

A.G. and M.M. designed the study. A.G. implemented the protocol and pipelines. A.G. and N.G. designed the evaluation scenario and wrote the paper. I.W. supervised the writing and provided feedback.

Competing Interests

The authors declare that no competing interests exist.

Additional information

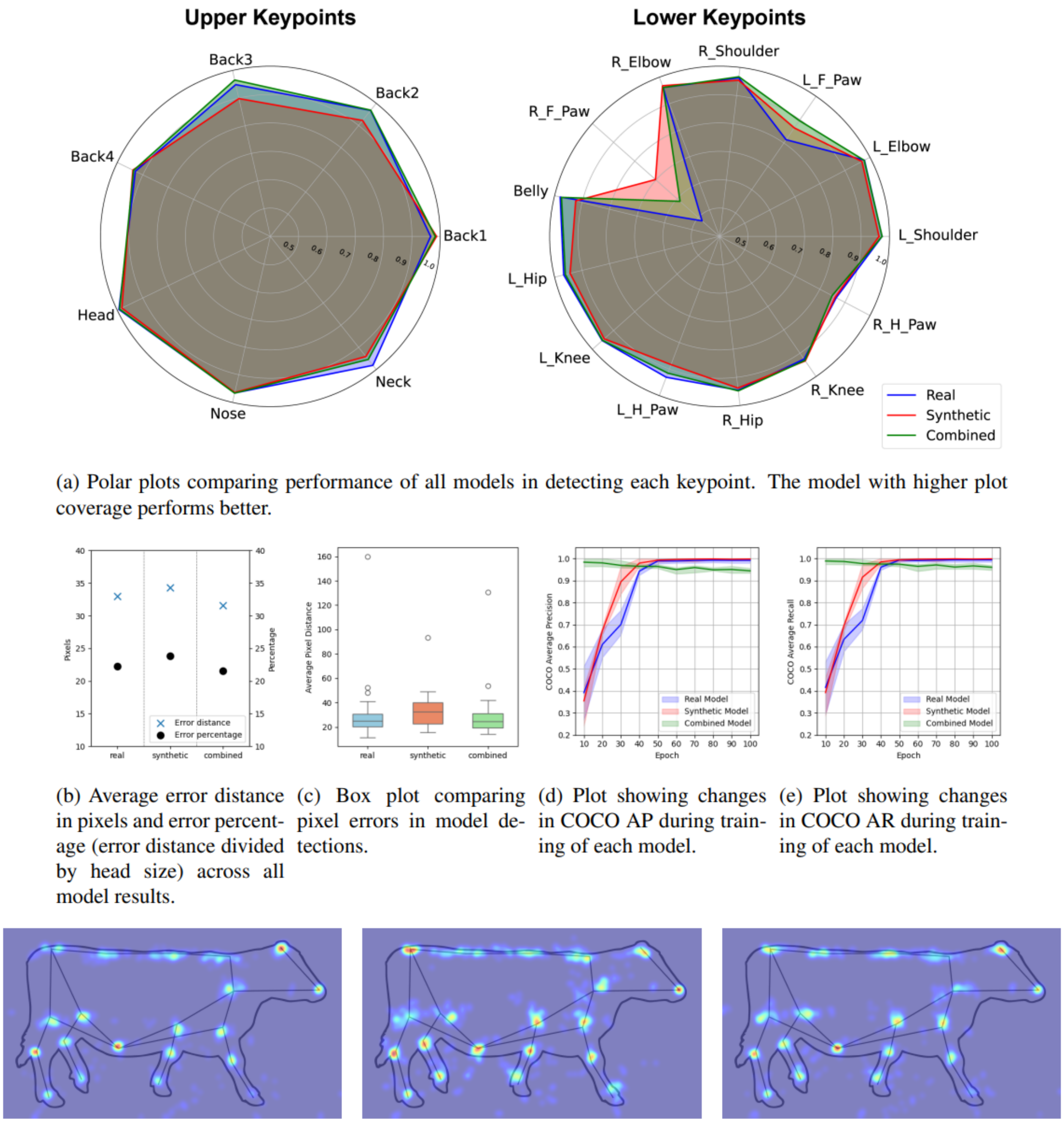
Supplementary materials can be found on the GitHub page of this project.

Table 1. Evaluation metrics showing the average performance of each model across a 5-fold cross validation with different values for intersection over union.

Metric	Real Model	Synthetic Model	Combined Model
COCO/AP @ IoU=0.50:0.95	0.86	0.87	0.87
COCO/AP @ IoU=0.75	0.90	0.90	0.88
COCO/AR @ IoU=0.50:0.95	0.89	0.89	0.91
COCO/AR @ IoU=0.75	0.93	0.93	0.93

Table 2. Description of datasets used for training each model

Model	Training Details
Real Model	99 Real Frames for Training and Validation
Synthetic Model	117 Generated Frames for Training and Validation
Combined Model	Real Model Fine-tuned on Synthetic Model’s Data, Real Model’s Validation



(a) Polar plots comparing performance of all models in detecting each keypoint. The model with higher plot coverage performs better.

(b) Average error distance in pixels and error percentage (error distance divided by head size) across all model results.

(c) Box plot comparing pixel errors in model detections.

(d) Plot showing changes in COCO AP during training of each model.

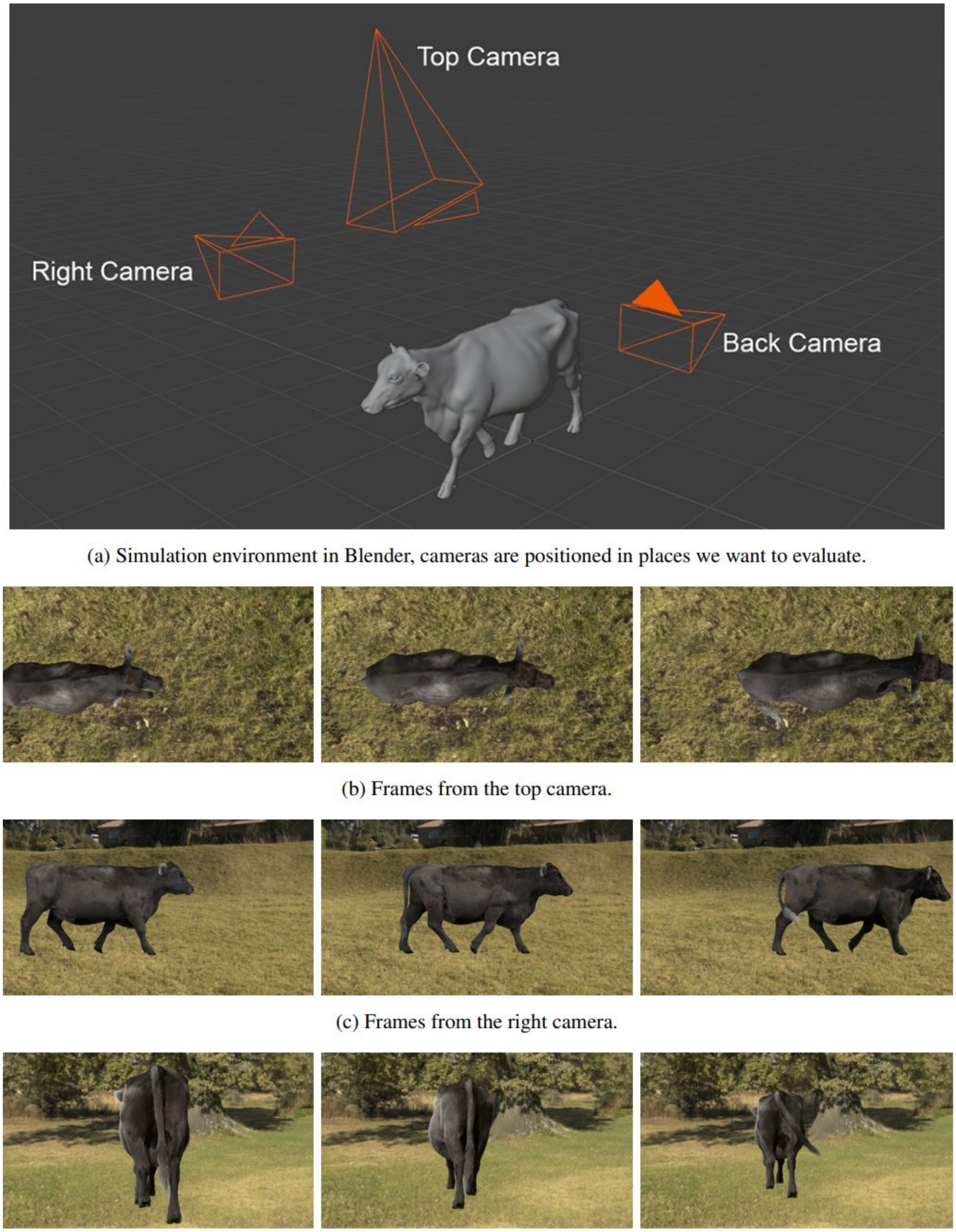
(e) Plot showing changes in COCO AR during training of each model.

(f) Heat map of predictions for the real model.

(g) Heat map of predictions for the synthetic model.

(h) Heat map of predictions for the combined model.

Figure 6. Plots comparing the performance of models. In pixel analysis, a 10-pixel length is equal to 2 cm on average.



(a) Simulation environment in Blender, cameras are positioned in places we want to evaluate.

(b) Frames from the top camera.

(c) Frames from the right camera.

(d) Frames from the back camera.

Figure 7. An example of a simulation process in Blender to specify the best location for placing a camera in real life scenario. Generated frames from each camera can be used to train deep learning models for the desired task. the model with the highest efficiency shows the best place to deploy the camera.