

1                   Using dorsal surface for individual  
2                   identification of dairy calves through 3D deep  
3                   learning algorithms

4                   Rafael E. P. Ferreira<sup>a</sup>, Tiago Bresolin<sup>a</sup>, Guilherme J. M.  
5                   Rosa<sup>a,b</sup>, João R. R. Dórea<sup>a,c,\*</sup>

6                   <sup>a</sup>Department of Animal and Dairy Sciences, <sup>b</sup>Department of Biostatistics  
7                   and Medical Informatics, <sup>c</sup>Department of Biological Systems Engineering,  
8                   University of Wisconsin-Madison, Madison-WI, USA 53706

9                   \*Corresponding author: [joao.dorea@wisc.edu](mailto:joao.dorea@wisc.edu)

---

10                  **Abstract**

11                  Advances in machine learning techniques have allowed the devel-  
12                  opment of computer vision systems (CVS) that can accurately  
13                  predict several phenotypes of interest for livestock operations.  
14                  In this context, 3D images taken from a top-down view are par-  
15                  ticularly useful for estimating body condition score, growth de-  
16                  velopment, and body biometrics in cattle. Frequently, such CVS  
17                  rely on identification (ID) systems, such as electronic tags, as  
18                  a way to match animal ID and the predicted phenotype. How-  
19                  ever, the same 3D images used to predict body weight and other  
20                  animal biometrics could be adopted also for animal recognition.  
21                  Such alternative would optimize CVS to recognize animal ID  
22                  and monitor growth development simultaneously, and leverage  
23                  the same hardware infrastructure. Furthermore, this strategy  
24                  could be used to recognize animals with similar color patterns.  
25                  Nonetheless, growing animals are continuously changing body  
26                  shape, which could limit its use as an invariant feature for pat-  
27                  tern recognition. Thus, the objectives of this study were: (1)  
28                  to compare algorithms for different 3D object representations  
29                  to identify individual animals; and (2) to evaluate how short-

*Preprint submitted to Computer and Electronics in Agriculture May 18, 2022*

30 term changes in body shape due to animal growth affect the  
31 predictive performance of these algorithms. For objective 1, the  
32 algorithms were trained ( $n = 4,558$ ) and tested ( $n = 1,139$ ) us-  
33 ing images from 38 Holstein calves. For objective 2, we designed  
34 three different experiments using images ( $n = 2,347$ ) from five  
35 Holstein calves taken over six weeks during their growing pe-  
36 riod, always training and testing on different weeks. Each ex-  
37 periment evaluated how changing a different parameter of the  
38 image capturing procedure affected the predictive ability of the  
39 trained algorithms. In the first experiment, we varied the total  
40 number of images per animal in the training set; in the second  
41 experiment, we varied the number of weeks while keeping a fixed  
42 number of images in the training set; and in the third experi-  
43 ment, we skipped weeks between images in the training and test  
44 sets. The  $F_1$  score for objective (1) was up to 0.804 when testing  
45 with the last frames of each video, and up to 0.959 when using  
46 random frames for testing. For objective (2), the  $F_1$  score was  
47 up to 0.947 for the first experiment when using 130 images per  
48 animal; up to 0.979 for the second experiment when using all  
49 five weeks; and up to 0.917 when not skipping weeks between  
50 training and testing. These results show that deep learning algo-  
51 rithms can be used to identify individual animals through their  
52 dorsal area 3D surface, and, from our experiments using calves  
53 in their growing period, that they are robust enough to account  
54 for changes in body shape and size, making them a promising  
55 tool for animal recognition during growth.

56 *Keywords:* Growth, Calves, Animal Traceability, Deep  
57 Learning, Animal Identification, 3D Neural Networks

---

## 58 **1. Introduction**

59 Deep learning techniques have gained great popularity in  
60 the field of computer vision in recent years due to their im-  
61 pressive performance in tasks such as image classification, ob-  
62 ject detection, and semantic segmentation (Voulodimos et al.,  
63 2018). Deep learning allows machine learning models to learn  
64 abstract feature representations of the input data and perform  
65 automatic feature extraction when exposed to large amounts of  
66 data (LeCun et al., 2015). Such advances in deep learning and  
67 computer vision, and particularly in the use of 3D cameras, have  
68 enabled the development of systems that capture animal pheno-  
69 types such as body condition, body weight, lameness, behavior  
70 traits, and more (Fernandes et al., 2020). In order to capture and  
71 use animal-level phenotypes, implementing a system to identify  
72 individual animals is vital. These systems can be manual, such  
73 as ear tags, or automated, such as radio-frequency identification  
74 (RFID) (Voulodimos et al., 2010). However, implementing man-  
75 ual identification or RFID systems in large scale operations can  
76 be labor-intensive, prone to human error and fraud, costly, and  
77 invasive for the animals, as it requires manually placing RFID  
78 tags on each animal.

79 In this context, using computer vision techniques to imple-  
80 ment both animal identification and phenotyping into one single  
81 integrated system can be beneficial, as it could limit the use of  
82 external accessories attached to animals, leverage the same hard-  
83 ware infrastructure, and therefore address most of the issues  
84 related to RFID systems. Moreover, CVS could be a robust al-  
85 ternative to track animals along the food supply chain, allowing  
86 the development of traceability programs with high degree of  
87 security as found in blockchain systems (Casino et al., 2019).

88 Recent studies have proposed the use of RGB (Red, Green,  
89 Blue) images to identify animals based on their unique coat  
90 color patterns in different species by using 2D (2-dimensional)  
91 convolutional neural networks (CNNs). Andrew et al. (2017)  
92 and Bello et al. (2020) used 2D CNNs to identify Holstein cows  
93 using top-view images of their back, Yao et al. (2019) used detec-  
94 tion and classification 2D CNNs to detect and identify Holstein  
95 cows using images of their face, Yukun et al. (2019) used RGB  
96 and depth images to automatically identify Holstein cows and  
97 estimate their body condition score, and Hansen et al. (2018)  
98 proposed their own 2D CNN to individually identify pigs using  
99 images of their face. However, these approaches require that in-  
100 dividual animals have different coat color patterns, so that they  
101 would likely fail to differentiate animals with similar colors pat-  
102 terns, or certain animal breeds that have little color distinction  
103 between individuals.

104 As an alternative to RGB images, different 3D data represen-  
105 tations can be used to classify objects. For example, depth im-  
106 ages, despite being virtual representations of 3D (3-dimensional)  
107 surfaces, can be used along with 2D CNNs to perform classifica-  
108 tion tasks, because they are actually 2D images where each pixel  
109 contains a value representing the distance between the physical  
110 point at that pixel and the camera sensor. Additionally, 3D  
111 CNNs and other neural network architectures have been recently  
112 proposed to work with other 3D representations, such as vox-  
113 els (Maturana and Scherer, 2015), octrees (Wang et al., 2017),  
114 and point clouds (Qi et al., 2016). These representations can  
115 prove beneficial in classifying objects whose 3D shape is more  
116 relevant than their color, as showed by Aijazi et al. (2013) when  
117 segmenting urban scenes, and Soilán Rodríguez et al. (2019)

118 when classifying data acquired with Airborne Laser Scanning  
119 systems, for example. Such tasks, however, can be challenging  
120 when working with objects that quickly change their shape over  
121 time, such as animals during their growing stage of life.

122 The current study aims to evaluate the predictive ability of  
123 deep neural networks to identify individual calves based on the  
124 shape of their dorsal region, using different 3D representations  
125 as input data. Additionally, we evaluated the robustness of the  
126 tested algorithms to perform this task as body shape changes  
127 due to animal growth. To accomplish that, we (1) compared  
128 algorithms for different 3D object representations to identify in-  
129 dividual animals by using images collected in the same period of  
130 time; and (2) evaluated how short-term changes in body shape  
131 due to animal growth affect the predictive performance of these  
132 algorithms.

## 133 **2. Materials and Methods**

134 This study was split into two objectives, as previously men-  
135 tioned. For the first one, we compared the performance of five  
136 neural network architectures on identifying individual calves by  
137 using different 3D data representations. Three of them were 2D  
138 CNNs using depth images as inputs, and the other two were a  
139 3D CNN using voxels as inputs, and a combination of multi-  
140 layer perceptrons using point clouds as inputs, respectively. For  
141 the second objective, the same five neural network architectures  
142 were assessed on identifying individual calves in different peri-  
143 ods of time during their growing stage, in order to evaluate how  
144 changes in body shape would affect the predictive performance  
145 of these algorithms.

146 *2.1. Datasets*

147 For the first objective, videos from 38 pre-weaned Holstein  
148 dairy calves with ages varying from two to eight weeks, and  
149 body weight (BW) of  $57.0 \pm 14.7$  kg (*average  $\pm$  SD*), housed at  
150 the Emmons Blaine Dairy Cattle Research Center (Arlington,  
151 WI), were recorded during a single week. A Kinect V2 sensor  
152 (Microsoft; Redmond, WA) was used, which has an RGB camera  
153 (resolution of  $1920 \times 1080$  pixels), a depth sensor (resolution of  
154  $512 \times 424$  pixels), and a microphone array. The 38 videos were  
155 recorded from a top-down view, and each contained a single calf,  
156 as they were recorded separately while weighing each animal  
157 individually. All videos were recorded using Kinect for Windows  
158 SDK 2.0 (Microsoft; Redmond, WA) installed on a laptop locally  
159 operated by a person who manually started recording as soon as  
160 the calf was positioned on the scale, and stopped recording when  
161 the weighing process was concluded for that calf. The length  
162 of the videos varied from 15 to 69 seconds, from which frames  
163 from the depth stream were extracted at a rate of four frames  
164 per second (FPS). This resulted in a total of 5,764 depth frames  
165 with a resolution of  $512 \times 424$  pixels, each pixel representing the  
166 distance from the object to the camera sensor in millimeters.

167 For the second objective, 30 videos from five calves with  
168 ages varying from four to eight weeks, and BW of  $63.8 \pm 6.7$  kg  
169 (*average  $\pm$  SD*), housed at the Dairy Cattle Research Center  
170 (DCRC; Madison, WI), were recorded using the same Kinect  
171 V2 sensor (Microsoft; Redmond, WA) from a top view, and the  
172 same recording procedures as in the first objective. Each calf  
173 had the videos recorded separately once a week for six weeks,  
174 with video recording lengths between 18 and 80 seconds. Depth  
175 frames were then extracted at a rate of two FPS, resulting in a

176 total of 2,347 frames with a resolution of  $512 \times 424$  pixels, each  
177 pixel representing the distance from the object to the camera  
178 sensor in millimeters.

## 179 *2.2. Data preprocessing*

180 Data preprocessing was performed for each acquired frame  
181 in each dataset, and it involved four steps, in the following or-  
182 der: background removal, point cloud generation, point cloud  
183 augmentation, and occupancy grid generation. The four steps  
184 are described in the following sub-sections.

### 185 *2.2.1. Background removal*

186 In order to remove background pixels from the captured  
187 depth images, a network based on the Mask R-CNN framework  
188 (He et al., 2018) was implemented to automatically detect and  
189 retain all pixels containing a calf. We only considered as part of  
190 the calf the region between the tail and the neck of the animal.  
191 The Mask R-CNN network was trained using 584 depth images  
192 manually segmented according to this standard, as shown in Fig-  
193 ure 2b, where pixels containing the calf appear in white. Some  
194 of the frames captured from the original videos did not contain  
195 a calf, resulting in 5,697 frames for the first objective, and 2,295  
196 for the second. The intersection over union of the trained net-  
197 work for image segmentation was 0.932 using an independent  
198 test set.

### 199 *2.2.2. Point cloud generation*

The pixels detected as containing a calf were converted to  
a set of points in a 3-dimensional coordinate system (a point  
cloud). For each pixel  $(i, j)$  containing a depth value  $d$ , a point  
 $(x_p, y_p, z_p)$  was created with values  $(x_p, y_p, z_p) = (j, i, d)$ . This

resulted in a point cloud with the number of points equal to the number of pixels that were part of a calf in the original frame. Outlier points were then removed based on their Z-axis coordinates, or depth value, in order to prevent the inclusion of background pixels due to segmentation errors. A value was considered an outlier if it was more than three scaled median absolute deviations (MAD) from the median. For a random vector  $\mathbf{X}$  with  $N$  scalar observations, the MAD is defined as follows:

$$MAD = \text{median}(|X_i - \text{median}(X)|) \quad (1)$$

200 for  $i = 1, 2, \dots, N$

201 The scaled MAD is defined as  $k \cdot MAD$ , where  $k \approx 1.4826$  is a  
 202 constant scale factor that depends on the distribution (Rousseeuw  
 203 and Croux, 1993). In this case, we are assuming the Z-axis val-  
 204 ues are normally distributed.

### 205 *2.2.3. Point cloud augmentation*

206 The generated point cloud was then augmented by randomly  
 207 rotating, scaling, and applying jitter to the point coordinates.  
 208 Image augmentation is a technique to avoid overfitting and add  
 209 robustness to 2D convolutional networks (Perez and Wang, 2017),  
 210 and point cloud augmentation is a similar technique with some  
 211 important differences. The main difference is in the rotation pro-  
 212 cess: point cloud augmentation allows the objects to be rotated  
 213 around any of the three axes, as opposed to image augmenta-  
 214 tion, where the image can only be rotated around a single axis.  
 215 In this study, the point clouds were rotated around their Z-axis  
 216 by a random angle between 0 and 360 degrees, the coordinates  
 217 were scaled by a random factor between 0.98 and 1.02, and a  
 218 1% jitter was applied to each point. These values were chosen

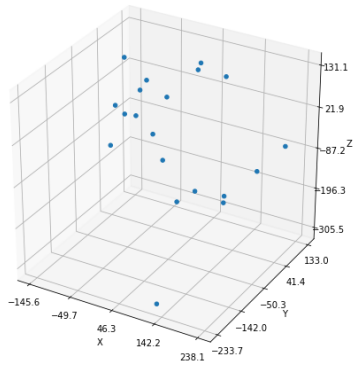
219 arbitrarily. Applying these transformations introduced noise to  
 220 the data, avoiding overfitting and making the trained models  
 221 more robust to rotation.

#### 222 2.2.4. Occupancy grid generation

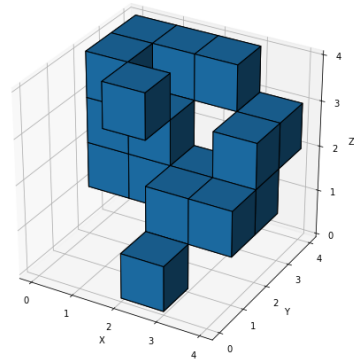
The point cloud resulting from the augmentation step was then converted to an occupancy grid, splitting the points' coordinate space into 32 cells on each axis. For each point  $(x_p, y_p, z_p)$ , the coordinate values of the containing cell in the grid space  $(x_{cell}, y_{cell}, z_{cell})$  were calculated as follows:

$$\begin{aligned}
 x_{cell} &= \min\left(\left\lfloor \frac{x_p - \min_x}{\max_x - \min_x} \cdot 32 \right\rfloor, 31\right) \\
 y_{cell} &= \min\left(\left\lfloor \frac{y_p - \min_y}{\max_y - \min_y} \cdot 32 \right\rfloor, 31\right) \\
 z_{cell} &= \min\left(\left\lfloor \frac{z_p - \min_z}{\max_z - \min_z} \cdot 32 \right\rfloor, 31\right)
 \end{aligned} \tag{2}$$

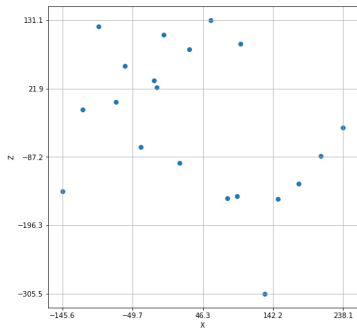
223 Values  $\min_x$  and  $\max_x$  were the minimum and maximum  $x_p$   
 224 values in the point cloud, and likewise for  $y$  and  $z$ , resulting in  
 225 values in the range  $[0, 31]$  for each cell coordinate. Based on the  
 226 cell coordinates of each point, the  $32 \times 32 \times 32$  grid was then filled  
 227 with ones or zeros depending on whether the corresponding cell  
 228 contained at least one point of the original point cloud (Figure  
 229 1). Occupancy grids can serve as a more regular 3D represen-  
 230 tation of the data in comparison to point clouds, with grid cells  
 231 contained in a discrete domain as opposed to the continuous na-  
 232 ture of point coordinates in point clouds. Such regularization  
 233 can help machine learning systems learn more efficiently than  
 234 with more irregular formats such as raw point clouds, by adopt-  
 235 ing 3D convolutional neural networks, for example (Maturana  
 236 and Scherer, 2015).



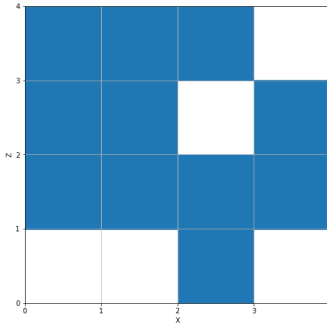
(a) Point cloud in 3D space



(b) Occupancy grid generated in a  $4 \times 4 \times 4$  grid space



(c) Point cloud projected on the XZ-plane



(d) Occupancy grid projected on the XZ-plane

Figure 1: Example of the occupancy grid generation process. (c) shows a point cloud in 3D space, (b) shows the corresponding generated occupancy grid in a  $4 \times 4 \times 4$  grid space, (c) shows the same point cloud projected on the XZ-plane, and (d) shows the corresponding occupancy grid projected on the XZ-plane. In the occupancy grids (b and d), filled cells are assigned value 1, and empty cells are assigned value 0. Examples were given in 3D and 2D for clarification.

237 Figure 2 shows an example of the step-by-step process of  
 238 transforming a depth frame into an occupancy grid.

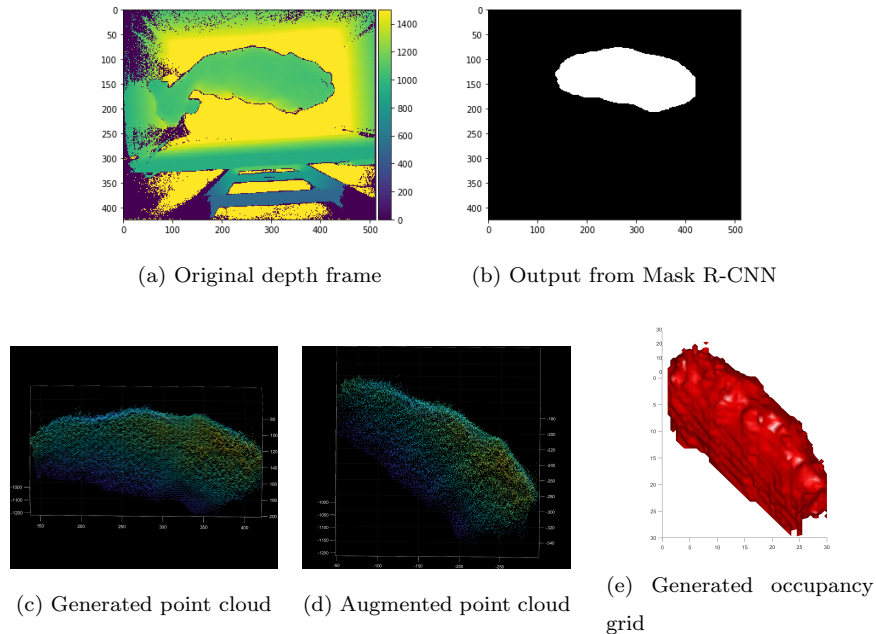


Figure 2: An example of all preprocessing stages applied to a depth image to generate an occupancy grid. A depth frame (a) is extracted from a video captured using the Kinect V2 sensor; a Mask R-CNN network detects the pixels containing the calf body, and generates a binary mask (b); this binary mask is applied to the point cloud generated from the depth frame, resulting in a point cloud of the calf body (c); this point cloud is then augmented (d) and used to generate the final occupancy grid (e).

239 *2.3. Training and test sets*

240 For the first objective, two different approaches were used to  
 241 split the dataset into training and test sets. In the first approach,  
 242 5,697 frames were randomly split into training ( $n = 4,558$ ) and  
 243 test ( $n = 1,139$ ) sets, corresponding to 80% and 20% of the to-  
 244 tal dataset, respectively, without necessarily maintaining class  
 245 proportions between training and test sets. This process was  
 246 repeated 10 times, generating 10 different random dataset splits  
 247 that were used to calculate an average final performance metric.  
 248 The randomization was done at the level of the entire 38 videos,  
 249 generating slightly different class proportions for each permuta-  
 250 tion. In the second approach, the frames from each video were

251 split chronologically based on their position in the video, sepa-  
 252 rating the first 80% frames for training and last 20% for testing.  
 253 We used the second approach to minimize similarities between  
 254 the training and test sets, as adjacent frames tend to be similar  
 255 to each other.

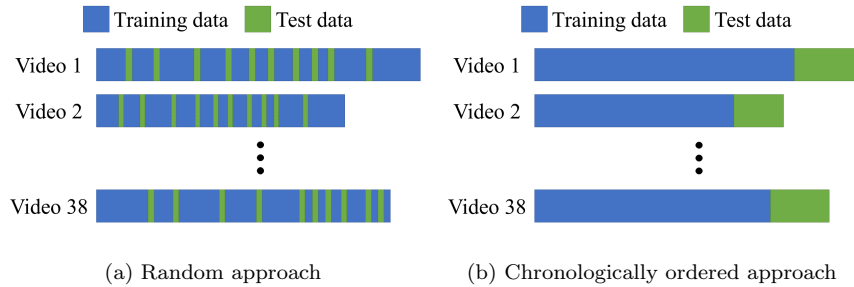


Figure 3: Dataset splits for the first objective. In the random approach (a), the dataset was split into training and test sets, including 80% and 20% of the frames, respectively. In the chronologically ordered approach (b), the frames from each video were assigned to the training or test sets based on their position in the video: the first 80% frames were assigned to the training set, and the last 20% were assigned to the test set.

256 For the second objective, three different experiments were  
 257 designed, and the dataset was split accordingly. The first exper-  
 258 iment consisted of evaluating how the number of frames used  
 259 in training would affect the predictive performance of the algo-  
 260 rithms. For that, random samples of 20, 40, 70, 100, 130, and  
 261 154 images per animal were used for training, all from the first  
 262 and second weeks, and a fixed set of 319 images from the third  
 263 week was used for testing.

264 In the second experiment, we evaluated how increasing the  
 265 number of consecutive weeks used for training affected the per-  
 266 formance of the algorithms on the immediate following week,  
 267 while keeping the same total amount of images per animal. We  
 268 used 80 images per animal for training (resulting in a total of

269 400 images), and tested on images from the following week, such  
 270 that the size of the test set varied according to the week, but the  
 271 training set size remained constant. A total of ten dataset splits  
 272 were created for this experiment, grouping them according to  
 273 the total number of weeks used for training, and calculating an  
 274 average performance for each group (Table 1).

<b>Group</b>	<b>Weeks in training set</b>	<b>Test week</b>	<b>Test set size</b>
Two weeks	1 and 2	3	319
	2 and 3	4	254
	3 and 4	5	250
	4 and 5	6	403
Three weeks	1, 2, and 3	4	254
	2, 3, and 4	5	250
	3, 4, and 5	6	403
Four weeks	1, 2, 3, and 4	5	250
	2, 3, 4, and 5	6	403
Five weeks	1, 2, 3, 4, and 5	6	403

Table 1: Splits performed for second objective, experiment 2. The ten splits were grouped according to the number of weeks used for training, and the four resulting groups were compared to evaluate the effect of adding more weeks to the training set.

275 Finally, in the third experiment, we evaluated the effect of  
 276 increasing the time interval between the training and test sets on  
 277 the prediction quality of the tested algorithms. In this context,  
 278 we defined four time intervals in relation to weeks after training:  
 279 zero (testing on images from the subsequent week), one, two, and  
 280 three weeks. For training, we used two consecutive weeks and  
 281 80 images per animal (resulting in a total of 400 images) for  
 282 each split. Ten splits were created for this experiment, grouped

283 according to the interval between the training and test sets. The  
 284 size of the test set varied according to the week used for testing  
 285 (Table 2).

<b>Group</b>	<b>Weeks in training set</b>	<b>Test week</b>	<b>Test set size</b>
No skipping	1 and 2	3	319
	2 and 3	4	254
	3 and 4	5	250
	4 and 5	6	403
Skipping one week	1 and 2	4	254
	2 and 3	5	250
	3 and 4	6	403
Skipping two weeks	1 and 2	5	250
	2 and 3	6	403
Skipping three weeks	1 and 2	6	403

Table 2: Splits performed for second objective, experiment 3. The ten splits were grouped according to the time interval between training and test sets, and the four resulting groups were compared to evaluate the effect of skipping weeks between training and testing.

286 Table 3 provides an overview of the three experiments per-  
 287 formed for the second objective.

<b>Experiment</b>	<b>Images per animal</b>	<b>Number of weeks</b>	<b>Time interval</b>
1	Varying	2	No skipping
2	80	Varying	No skipping
3	80	2	Varying

Table 3: Experiments performed for the second objective. The experiments evaluated how changing the number of images per animal, number of weeks used for training, and time interval between training and testing affected the predictive performance of the algorithms.

288 *2.4. Data representations and algorithms*

289 The algorithms were chosen based on the data representa-  
290 tion used as input. Algorithms able to analyze 2D depth images  
291 (Simonyan and Zisserman, 2014; Szegedy et al., 2016; Chollet,  
292 2017), point clouds (Qi et al., 2016), and occupancy grids (Mat-  
293 urana and Scherer, 2015) were selected. All algorithms were im-  
294 plemented in Python, using TensorFlow (Abadi et al., 2015) for  
295 implementing PointNet, TensorFlow and Keras (Chollet et al.,  
296 2015) for implementing VGG16, Inception v3 and Xception, and  
297 Theano (Theano Development Team, 2016) and Lasagne (Diele-  
298 man et al., 2015) for implementing VoxNet.

299 *2.4.1. Depth images – VGG16, Inception v3, and Xception*

300 To generate depth images from the extracted video frames,  
301 the data was processed using only the first preprocessing stage,  
302 described in Section 2.2.1. The resulting mask was applied to  
303 the pixel-based depth values, setting every pixel not contained in  
304 the mask to zero. Outliers were then identified using the method  
305 presented in Section 2.2.2, and their corresponding values were  
306 set to zero. The final depth image consisted of a matrix of size  
307  $424 \times 512$  containing the depth values of relevant pixels, or zero  
308 for pixels considered part of the background.

309 These depth images were then used as the input to three  
310 different deep neural network (DNN) architectures: VGG16 (Si-  
311 monyan and Zisserman, 2014), Inception v3 (Szegedy et al.,  
312 2016), and Xception (Chollet, 2017). For all three DNNs, the  
313 last Fully-Connected (FC) layer of the original architecture was  
314 removed, and all the other layers were initialized with weights  
315 from the respective networks trained using ImageNet (Deng et al.,  
316 2009), an open image dataset containing more than 1 million ex-

317 am-  
318 ples of diverse objects and environments, ranging from wild  
319 and farm animals to vehicles, airplanes, and housewares, for  
320 example. Such strategy was defined by Weiss et al. (2016) as  
321 Transfer Learning, and it accelerates the training process as the  
322 network weights are initialized with values optimized for a large  
323 generic image dataset such as ImageNet, instead of being ini-  
324 tialized with random values. This technique helped our new  
325 networks learn generic features, such as textures, edges, cor-  
326 ners, and shapes, previously learned in a different task domain  
using a much larger dataset.

327 The VGG16-based network was extended with a FC layer of  
328 size 2048 and a Rectified Linear Unit (ReLU) activation function  
329 (Nair and Hinton, 2010), followed by a final FC layer of size  $n$   
330 and softmax activation function, where  $n$  is the number of classes  
331 for each objective ( $n = 38$  for the first objective and  $n = 5$  for  
332 the second objective).

333 The Inception v3- and Xception-based DNNs were extended  
334 with a global average pooling layer as described by Lin et al.  
335 (2014), followed by a FC layer of size 1024 and ReLU activation  
336 function, and a final FC layer of size  $n$  and softmax activation  
337 function, similarly to the VGG16-based approach.

338 For each DNN, the training process was split into two consec-  
339 utive stages: feature extraction and fine-tuning. In the feature  
340 extraction stage, the DNN was trained for 200 epochs keeping  
341 the weights of all but the last two FC layers frozen. This al-  
342 lowed features previously learned through Transfer Learning to  
343 be used and retained. In the fine-tuning stage, weights from ear-  
344 lier layers were unfrozen, and the network was trained for 400  
345 epochs with a smaller learning rate, allowing it to further learn  
346 features that are more specific to our context.

347 The VGG16-based network was trained using RMSProp (Hin-  
348 ton et al., 2012) with a learning rate of  $2 \times 10^{-5}$  in the feature  
349 extraction stage and  $1 \times 10^{-5}$  in the fine-tuning stage. The In-  
350 ception v3-based network was trained using RMSProp with a  
351 learning rate of  $1 \times 10^{-3}$  in the feature extraction stage, and  
352 Stochastic Gradient Descent (Robbins and Monro, 1951) with a  
353 learning rate of  $1 \times 10^{-4}$  and momentum of 0.9 (Qian, 1999) in  
354 the fine-tuning stage. The Xception-based network was trained  
355 using Adam (Kingma and Ba, 2014) with a learning rate of  
356  $1 \times 10^{-3}$  in the feature extraction stage and  $1 \times 10^{-5}$  in the  
357 fine-tuning stage.

#### 358 *2.4.2. Point cloud – PointNet*

359 From the point clouds generated by applying the first three  
360 preprocessing stages described in Section 2.2, the k-means clus-  
361 tering algorithm was used to separate the 3D points into 2,048  
362 clusters. The centroids of these clusters were then grouped into  
363 a new point cloud and used as the input to a network based on  
364 the full PointNet architecture (Qi et al., 2016). We decided to  
365 use point clouds of size 2,048 because PointNet was designed,  
366 trained, and validated using the ModelNet40 dataset (Wu et al.,  
367 2015), which contains point clouds of size 2,048. The last FC  
368 layer of the original PointNet architecture was modified to have  
369  $n$  nodes, where  $n$  is the number of classes for each objective,  
370 as before. The network was trained for 250 epochs using Adam  
371 with an initial learning rate of  $1 \times 10^{-3}$ , a momentum of 0.9,  
372 and exponential learning rate decay of 0.7 every 200,000 steps.

#### 373 *2.4.3. Occupancy grid (Voxel) – VoxNet*

374 The occupancy grids generated from applying all four pre-  
375 processing stages described in Section 2.2, also known as voxels,

376 were used as the input to a network based on the VoxNet archi-  
 377 tecture (Maturana and Scherer, 2015). The grid size was defined  
 378 as  $32 \times 32 \times 32$ , the same as proposed in the original VoxNet  
 379 article (Maturana and Scherer, 2015). The last FC layer of the  
 380 architecture was modified to have  $n$  nodes, the number of classes  
 381 for each objective. The network was trained for 400 epochs using  
 382 Stochastic Gradient Descent with a learning rate of  $1 \times 10^{-3}$ , a  
 383 momentum of 0.9, and  $L2$  norm regularization of 0.001 applied  
 384 to the loss function.

### 385 2.5. Evaluation metrics

386 To evaluate and compare the prediction quality of all algo-  
 387 rithms, the accuracy, precision, recall and  $F_1$  score were calcu-  
 388 lated for each class as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

$$F_1 = \frac{2TP}{2TP + FP + FN} \quad (6)$$

389 where:

390  $TP$  = True positive,  $TN$  = True negative,  $FP$  = False pos-  
 391 itive,  $FN$  = False negative

392 The mean values across all classes were then calculated and  
 393 used to compare the algorithms (macro averaging), and the final  
 394  $F_1$  score was calculated as the mean of the class-wise  $F_1$  scores.  
 395 Precision, recall, and  $F_1$  score are important metrics to evaluate  
 396 classification tasks. They can be more informative than the  
 397 accuracy in a context of imbalanced data, where the number of  
 398 images corresponding to each class varies significantly.

### 399 3. Results and Discussion

#### 400 3.1. Comparing algorithms and 3D representations

401 The first objective consisted of comparing different algo-  
402 rithms and 3D representations to identify individual calves using  
403 their dorsal surface. The results discussed in this subsection are  
404 related to the approach of using a chronologically ordered split  
405 of the frames, in order to prevent overoptimistic results from  
406 using adjacent frames in the training and test sets (refer to Sec-  
407 tion 2.3 for details). Preventing biased evaluation results is an  
408 important step in any Artificial Intelligence system, as the main  
409 goal of the evaluation process is to try to anticipate how the  
410 algorithm will perform when facing real-world scenarios. Thus,  
411 when working with algorithms designed to generate predictions  
412 on images that will be captured in the future, it is critical to  
413 use the earliest captured images as the training set, and include  
414 only the latest captured images in the test set, in order to achieve  
415 more realistic results. A report of all calculated  $F_1$  scores can  
416 be found in Table 4. Using random images in a sequence for  
417 training and testing generates higher, overestimated  $F_1$  scores,  
418 when compared to a more realistic scenario of the test set con-  
419 taining only the last frames of the original videos. For example,  
420 when using Xception, the  $F_1$  score decreases from 0.959 to 0.804  
421 when using the chronological order approach, which is a more  
422 realistic approximation of how that network would perform on  
423 future images.

Train-test split	Data representation	Architecture	$F_1$ score
RO <sup>1</sup>	DI <sup>3</sup>	VGG16	0.888
RO <sup>1</sup>	DI <sup>3</sup>	Inception v3	0.904
<b>RO<sup>1</sup></b>	<b>DI<sup>3</sup></b>	<b>Xception</b>	<b>0.959</b>
RO <sup>1</sup>	PC <sup>4</sup>	PointNet	0.669
RO <sup>1</sup>	OG <sup>5</sup>	VoxNet	0.880
CO <sup>2</sup>	DI <sup>3</sup>	VGG16	0.718
CO <sup>2</sup>	DI <sup>3</sup>	Inception v3	0.750
<b>CO<sup>2</sup></b>	<b>DI<sup>3</sup></b>	<b>Xception</b>	<b>0.804</b>
CO <sup>2</sup>	PC <sup>4</sup>	PointNet	0.429
CO <sup>2</sup>	OG <sup>5</sup>	VoxNet	0.656

Table 4:  $F_1$  scores for each combination of train-test split, data representation, and network architecture for objective 1. The best performing network was the one based on the Xception 2D CNN architecture. <sup>1</sup>Random order; <sup>2</sup>Chronological order; <sup>3</sup>Depth images; <sup>4</sup>Point cloud; <sup>5</sup>Occupancy grid (voxel).

424 The 2D CNN approaches achieved  $F_1$  scores of 0.718, 0.750,  
425 and 0.804 with the VGG16-, Inception v3- and Xception-based  
426 networks, respectively. These results were consistent with the  
427 results reported in the original Xception publication (Chollet,  
428 2017), with Xception performing better than VGG16 and Incep-  
429 tion v3 on the ImageNet and JFT datasets. This improvement  
430 comes from making use of inception modules (Szegedy et al.,  
431 2015) and introducing depthwise separable convolutions (Chol-  
432 let, 2017).

433 The point cloud approach using a PointNet-based network  
434 achieved an  $F_1$  score of 0.429, which is the lowest of all the  
435 approaches for this objective. This is probably because, be-  
436 fore being fed to the network, the original point clouds resulted

437 from the preprocessing step in Section 2.2.3 were reduced from  
438 approximately 30,000 to 2,048 points. This downsampling was  
439 much stronger than the one performed in the original Point-  
440 Net article (Qi et al., 2016), which proposed a downsampling  
441 of the point clouds in the ModelNet40 dataset from 2,048 to  
442 1,024 points. This may have caused our network to miss impor-  
443 tant nuances from the surface of the calves, which are necessary  
444 to uniquely identify them. This evidence was supported when  
445 we tested this PointNet approach using only 1,024 points, and  
446 it resulted in a further  $F_1$  score drop to 0.318. The PointNet  
447 architecture was designed to recognize objects that are struc-  
448 turally very different from each other, such as cars, tables, and  
449 airplanes. When distinguishing such different objects, it is not  
450 significantly detrimental to make use of fewer points, because  
451 the network learns how to use a collection of critical points to  
452 summarize the shapes (Qi et al., 2016), and the summarized  
453 shapes are usually very different from each other. However, this  
454 architecture may not be suitable for objects that are very simi-  
455 lar in shape, and which the difference between individuals is in  
456 small details and nuances, such as in the case of identification  
457 of calves.

458 Using voxels as input, the VoxNet-based network achieved  
459 an  $F_1$  score of 0.656, which is superior to the results achieved  
460 using PointNet, but still below any of the  $F_1$  scores achieved us-  
461 ing 2D CNNs. VoxNet performed better than PointNet mostly  
462 because the voxels used in this study had a higher dimension-  
463 ality than the 2,048-sized point clouds. They were contained in  
464 grids of  $32 \times 32 \times 32$  cells, so a total of 32,768 cells each. How-  
465 ever, 2D CNNs performed better than VoxNet, possibly because  
466 they contain more parameters, allowing them to represent more

467 complex functions and to extract greater levels of details from  
468 the inputs. Extracting high-dimensional feature representations  
469 appears to be beneficial for individual calf recognition, as shown  
470 in the results. Additionally, the 2D CNNs used were pre-trained  
471 using the ImageNet dataset (Deng et al., 2009), as previously de-  
472 scribed in Section 2.4.1, which helped them learn more generic  
473 features before being trained with our datasets, further improv-  
474 ing their results in comparison to PointNet and VoxNet, which  
475 did not undergo any pre-training step. It is worth noting that  
476 other publicly available datasets could be used for pre-training  
477 the 2D CNNs, such as datasets containing exclusively images of  
478 animals, for example, which would be more similar to the input  
479 images used in this study. However, we could not find publicly  
480 available weights for Inception v3, VGG16, or Xception archi-  
481 tectures pre-trained using such animal datasets, and training  
482 those networks from scratch requires significant amount of time  
483 and computation resources, especially when using large image  
484 datasets (Simonyan and Zisserman, 2014; Szegedy et al., 2016;  
485 Chollet, 2017). Future research could be done to evaluate how  
486 the choice of pre-training dataset for transfer learning affects  
487 the predictive performance of neural networks for animal identi-  
488 fication, assessing the trade-off between using a dataset that is  
489 more similar to the one used in the final task, as opposed to a  
490 larger, more general dataset such as ImageNet.

491 Networks that contain more parameters, combined with higher-  
492 dimensional inputs, perform better in the task of calf identifi-  
493 cation using 3D images of their dorsal surface, as they can cap-  
494 ture more subtle variations in their shape. Such nuances can  
495 be helpful when trying to uniquely identify individuals. The  
496 depth images used in this study contained approximately 30,000

497 foreground pixels, the voxel grids contained 32,768 cells, and  
498 the point clouds used for PointNet contained only 2,048 points.  
499 The Xception-based network used had 23 million parameters,  
500 while the PointNet-based network had just 3.5 million, and the  
501 VoxNet-based network had less than 1 million. This possibly ex-  
502 plains why the Xception-based network was the best performing  
503 algorithm in this task when compared to point cloud- and voxel-  
504 based representations and architectures (PointNet and VoxNet),  
505 and these results agree with another work in the literature that  
506 performs similar comparisons for human face recognition (Pini  
507 et al., 2021).

508       Although 2D CNNs performed better in this specific setting  
509 where all videos were taken from a top-down view of the animals,  
510 2D depth images can only hold surface information about one  
511 specific view of an object. Conversely, 3D representations such  
512 as point cloud and voxel bring the possibility to merge multiple  
513 views of the same object into one single instance (Narayanan  
514 et al., 1998; Seitz et al., 2006), and hold volumetric informa-  
515 tion about an object. This enables deep learning algorithms to  
516 perform classification and identification tasks using multi-view  
517 3D representations, which contain a more robust and accurate  
518 depiction of the real object, possibly leading to better results  
519 (Gezawa et al., 2020). Although in this study we only used  
520 cameras positioned in a single fixed angle, it would be possible  
521 to take pictures from multiple different angles and build a full  
522 3D volumetric representation of the calves. Moreover, while 2D  
523 representations are limited to rotations around a single axis, 3D  
524 representations can be augmented by rotating the object around  
525 all three axes, or even by implementing an automated data aug-  
526 mentation policy, generating more realistic unseen versions of

527 the same animal (Cheng et al., 2020).

528       The networks employed in this study were trained using im-  
529 ages of the animals taken exclusively from a top-down view, and  
530 thus they can only effectively identify individual animals in new  
531 images taken from that same angle. Alternatively, if the exper-  
532 iment included images taken from different angles, it would be  
533 necessary to utilize a separate augmentation process for each  
534 group of depth images taken from the same angle. For example,  
535 if four synchronized cameras were positioned to take pictures of  
536 the same animal from different angles, they would generate four  
537 2D depth images per time point and animal, each undergoing a  
538 separate augmentation process. However, when using 3D repre-  
539 sentations such as voxels and point clouds, one single instance  
540 could represent the whole 3D animal by assembling images taken  
541 from different angles and reconstructing a full 3D model of the  
542 animal, as described by Narayanan et al. (1998), allowing for  
543 more effective augmentation approaches, such as the ones re-  
544 ported by Hahner et al. (2020) and Cheng et al. (2020). In this  
545 case, four pictures taken from synchronized cameras would re-  
546 sult in a single 3D voxel or point cloud. Such process could  
547 enhance the performance of the trained networks and yield su-  
548 perior results, as they could better generalize to a wider variety  
549 of camera angles and animal positions in this setting where im-  
550 ages are captured from different views simultaneously (Gezawa  
551 et al., 2020; Cheng et al., 2020; Hahner et al., 2020). In this situ-  
552 ation, 3D representations and networks could prove more useful  
553 than their 2D counterparts, despite achieving worse results in  
554 the context of our study.

555 *3.2. Evaluating how short-term changes in body shape affects the*  
556 *predictive performance of the algorithms*

557 Several situations can cause fast body shape changes in a  
558 short period of time, such as growth development in young ani-  
559 mals (Cominotte et al., 2020), or body tissue mobilization to  
560 supply energy demands in early lactating dairy cows (Dórea  
561 et al., 2017). Monitoring an animal throughout a long period of  
562 its life, including such periods of body shape change, can have se-  
563 rious implications in animal disease control and food traceability,  
564 by making it possible to backtrack disease outbreaks in a farm,  
565 and ensure that products derived from that animal follow local  
566 sanitary regulations (Awad, 2016). However, such changes could  
567 hinder the predictive performance of the evaluated algorithms,  
568 as an individual in an image captured in the future could look  
569 different from when previous images were captured and used for  
570 training the animal identification algorithms. Nevertheless, as  
571 is the case for human faces (Park et al., 2010), there might be  
572 unique biometric features and landmarks on the body shape of  
573 the animals that remain proportional and recognizable regard-  
574 less of the overall change in body size and shape. If the utilized  
575 algorithms are not robust enough to identify these features and  
576 account for body variations, they would have to be retrained  
577 frequently during these periods of intense body shape change.  
578 Frequently retraining such convolutional neural networks could  
579 be extremely costly and labor intensive, as they would require  
580 a large dataset of new labeled images (LeCun et al., 2015), and  
581 the labelling process would consist of manually assigning each  
582 image to the correct animal. Because of that, it is important to  
583 evaluate if the utilized algorithms can still accurately identify in-  
584 dividual animals even as they experience body changes. Thus,

585 the second objective of this study was to evaluate how short-  
586 term changes in body shape affects the predictive performance  
587 of the algorithms.

588 For the first experiment, which consisted of evaluating how  
589 the number of training images affected the predictive perfor-  
590 mance of the algorithms, the best results were achieved using  
591 the VoxNet-based network. Since we only used the first three  
592 weeks of data for this experiment, the simpler VoxNet architec-  
593 ture was sufficient, and the greater number of parameters and  
594 complexity of the Xception architecture did not translate into  
595 better results in this case. As shown in Table 5, increasing the  
596 number of training images per animal from 20 to 100 improved  
597 the  $F_1$  score from 0.734 to 0.944. This shows that deep neu-  
598 ral networks usually benefit from having more images available  
599 during training, so they can learn more intricate patterns and  
600 diverse examples from the training set, which help them better  
601 generalize to new data (LeCun et al., 2015). In our experiment,  
602 using more than 100 images per animal did not further improve  
603 the algorithms’ performance significantly, probably due to the  
604 uniformity of our dataset, with all images captured from the  
605 same view and location. Therefore, including more images pos-  
606 sibly just added more redundancy to the training set.

Images per animal	VGG16	Inception v3	Xception	PointNet	VoxNet
20	0.641	0.656	0.539	0.603	<b>0.734</b>
40	0.546	0.770	0.701	0.697	<b>0.917</b>
70	0.558	0.859	0.827	0.656	<b>0.929</b>
100	0.605	0.757	0.852	0.727	<b>0.944</b>
130	0.629	0.788	0.910	0.653	<b>0.947</b>
154	0.643	0.763	0.858	0.630	<b>0.939</b>

Table 5:  $F_1$  scores for each combination of images per animal and network architecture for the first experiment of objective 2. The VoxNet-based network achieved the best results in this experiment. Increasing the number of training images generally improved the  $F_1$  scores, up until 100 images per animal.

607 For the second experiment, which consisted of evaluating  
608 how the number of consecutive weeks used for training influ-  
609 enced the performance of the algorithms on the immediate next  
610 week, the best results were achieved using the Xception-based  
611 network. Table 6 shows that, even as the training set size re-  
612 mained constant, including more weeks slightly increased the  $F_1$   
613 score of this network, and the highest score improvement hap-  
614 pened when adding a fourth week to the training set. However,  
615 the PointNet- and VoxNet- based networks did not benefit from  
616 adding more weeks. This is likely because the Xception-based  
617 network, with a great number of parameters and trained using  
618 high resolution depth images, was the only network complex  
619 enough to capture useful information contained in more than  
620 two weeks concurrently. For the VoxNet-based network, with  
621 fewer parameters, and the PointNet-based network, using rela-  
622 tively low density point clouds, additional weeks possibly just  
623 translated into more noise added to the training set, not con-  
624 tributing for better results.

625 Nevertheless, these results show that even by using just two  
626 weeks, both VoxNet and Xception could learn sufficient pat-  
627 terns from the 3D shape of the calves to identify them on the  
628 next week. This means that it is not necessary to accumulate  
629 a long history of labeled images before being able to identify  
630 animals in new images, even if these animals are in a growing  
631 stage. According to the outcomes of our experiments, depth im-  
632 ages of the back of calves as young as three weeks old can be  
633 used to train networks able to identify them during the follow-  
634 ing week, showing that 3D deep learning systems can be used  
635 to monitor animals from a very early stage of life. Monitoring  
636 animals from an early stage is key for disease control as there is  
637 a high incidence of infectious diseases during that period (Marcé  
638 et al., 2010; Cho and Yoon, 2014). Thus, such identification and  
639 monitoring systems can help farmers make better management  
640 decisions to minimize the occurrence of such diseases and pre-  
641 vent the high economic losses associated with them (Kaneene  
642 and Hurd, 1990; Esslemont and Kossaibati, 1999).

Number of weeks	VGG16	Inception v3	Xception	PointNet	VoxNet
2	0.683	0.795	0.909	0.643	<b>0.911</b>
3	0.724	0.776	<b>0.906</b>	0.581	0.880
4	0.695	0.706	<b>0.970</b>	0.463	0.903
5	0.747	0.635	<b>0.979</b>	0.395	0.888

Table 6:  $F_1$  scores for each combination of number of weeks used for training and network architecture for the second experiment of objective 2. The Xception-based network achieved the best results in this experiment. The highest score improvement happened when adding a fourth week to the training set.

643 For the third experiment, which consisted of evaluating how  
644 skipping weeks between training and test sets affected the pre-

645 dictive performance of the algorithms, the best results were  
646 achieved, again, using the Xception-based network. Table 7  
647 shows how skipping weeks between training and testing affects  
648 the algorithms' predictive performance. Using the Xception-  
649 based network, the  $F_1$  score decreased from 0.917 to 0.846 when  
650 skipping one week. However, skipping more weeks did not fur-  
651 ther decrease the  $F_1$  score of this network considerably, showing  
652 that it might be possible to skip up to three weeks between  
653 training the network and identifying calves in new images with-  
654 out significantly affecting its predictive performance. This is  
655 evidence that the network might be learning unique biometric  
656 features on the body surfaces that remain proportional as the  
657 animals grow. Thus, although labeling new images and retrain-  
658 ing the network every week would yield the best results, it is still  
659 viable to train the network once and use it to identify calves on  
660 images taken three weeks later without a significant effect on  
661 the predictive ability.

662 By retraining the network only every three weeks, it is pos-  
663 sible to reduce the time and effort dedicated to labeling new  
664 images and performing the network training routine. Build-  
665 ing upon the previous experiment, depth images of the back of  
666 young calves can be used to train a network able to identify  
667 them during the three subsequent weeks, further improving the  
668 capacity of deep learning algorithms to monitor animals from  
669 an early stage of life. Such algorithms can contribute with the  
670 advancement of animal traceability and infectious diseases con-  
671 trol, ultimately improving farm productivity, food safety and  
672 consumer trust, and production sustainability (Awad, 2016).

Time interval	VGG16	Inception v3	Xception	PointNet	VoxNet
No skipping	0.704	0.746	<b>0.917</b>	0.533	0.917
1 week	0.595	0.612	<b>0.846</b>	0.551	0.831
2 weeks	0.535	0.654	<b>0.835</b>	0.441	0.806
3 weeks	0.753	0.726	<b>0.856</b>	0.282	0.792

Table 7:  $F_1$  scores for each combination of number of weeks skipped between training and testing and network architecture, for the third experiment of objective 2. The Xception-based network achieved the best results in this experiment. Skipping one week affected the  $F_1$  score, but it remained constant after further skipping more weeks.

673 Deep learning algorithms can be used to identify individual  
674 animals using their dorsal area 3D surface and, based on our ex-  
675 periments using calves in their growing period, they are robust  
676 enough to account for changes in body shape and size of the  
677 same animals. This study focused on calves in their early stage  
678 of life because that is when they undergo the most significant  
679 changes in body shape and size, representing a more challenging  
680 setting for machine learning algorithms, as significant divergence  
681 between training and future (or testing) data distributions of-  
682 ten hinder such algorithms’ predictive performance. Conversely,  
683 when working with mature cows that show a more limited body  
684 shape variability, the training data distribution would be more  
685 similar to that of images collected in the future (images of inter-  
686 est for identification), thus representing a less challenging setting  
687 for machine learning algorithms. In fact, Andrew et al. (2016)  
688 and Okura et al. (2019) used RGB-D images to identify ma-  
689 ture Holstein dairy cows. Nevertheless, adult dairy cows can  
690 still undergo significant body shape changes during the tran-  
691 sition period (between late pregnancy and early lactation), as  
692 they mobilize fat stores to compensate for a high milk yield and

693 relatively low dry matter intake. Thus, although not explicitly  
694 shown in this study, algorithms that are able to identify individ-  
695 ual calves as their body shapes change have the potential to be  
696 useful for monitoring mature dairy cows during their transition  
697 period, and future studies could explore this possibility.

698        Depending on the task complexity, 2D CNNs with a higher  
699 representation capacity, as a consequence of having a greater  
700 number of parameters, can achieve better results than their  
701 3D counterparts on identifying individual animals as their body  
702 grow. Nevertheless, regardless of the representation approach,  
703 3D information can be used in computer vision systems that  
704 identify individual animals based exclusively on their shape, in-  
705 stead of relying on coat color pattern information. Methods  
706 that rely on unique color patterns, such as the ones proposed  
707 by Andrew et al. (2017), Bello et al. (2020), Yao et al. (2019),  
708 Yukun et al. (2019), and Hansen et al. (2018), are limited to  
709 only certain animal breeds, in scenario with no significant body  
710 occlusion. Alternatively, deep learning methods that use solely  
711 3D information for individual identification can potentially be  
712 applied on species and breeds that share similar color patterns  
713 across individuals, such as Jersey, Brown Swiss, and Angus cat-  
714 tle, Rambouillet sheep, Saanen goats, Yorkshire pigs, and oth-  
715 ers; and in production systems where animals can be covered in  
716 mud or dirt, such as free range systems for pigs. By enabling  
717 the use of animal biometrics to perform individual identifica-  
718 tion in a multitude of species, breeds and production systems,  
719 these 3D deep learning algorithms push the boundaries of ani-  
720 mal traceability and phenotyping. Although Yukun et al. (2019)  
721 makes use of depth and RGB images to perform animal iden-  
722 tification, to the best of our knowledge this is the first work

723 to propose the exclusive use of depth images and 3D represen-  
724 tations for individual animal identification through 2D and 3D  
725 CNNs. Furthermore, this is also the first study to evaluate the  
726 ability of convolutional neural networks to identify animals as  
727 they grow rapidly and experience intense body shape changes.  
728 Moreover, since they are based on animal biometrics that can-  
729 not be easily manipulated by humans, the methods proposed in  
730 this work provide a secure and automated way of tracking indi-  
731 vidual animals along the food supply chain, contributing as an  
732 additional tool for ensuring food safety to consumers.

733 As previously mentioned, although deep learning methods  
734 represent the state-of-the-art in many computer vision appli-  
735 cations, they often require large amounts of training data to  
736 efficiently learn a certain task. This could pose an obstacle  
737 to commercial applications where labeled data is not so read-  
738 ily available. In that context, implementing hybrid approaches  
739 that merge traditional computer vision techniques with deep  
740 learning might help reduce the need for labeled data and de-  
741 crease training times (O’Mahony et al., 2019). Alternatively,  
742 active learning techniques (Settles, 2009) can be used to include  
743 human input in the learning process to optimize data annota-  
744 tion (for example, the system could request more examples of  
745 cows that are harder to classify or classes that are underrep-  
746 resented). In addition, semi-supervised learning methods can  
747 leverage information contained in both labeled and unlabeled  
748 data to build high-performing classifiers, potentially requiring  
749 smaller amounts of labeled data for training (Zhu, 2005).

750 In future research, capturing images during longer periods  
751 of time throughout the animals’ life might help understand how  
752 long a trained network can still be useful for individual recog-

753 nition without the need to retrain it. Additionally, it would be  
754 interesting to explore how the proposed methods would apply to  
755 mature dairy cows and other animal species and breeds such as  
756 Angus cows, or Yorkshire pigs. It would be beneficial to include  
757 more individuals in a future study as well, bringing the context  
758 closer to that of a commercial farm. In fact, commercial farms  
759 rarely hold a fixed herd for a long time. Instead, animals are con-  
760 stantly added or removed from the herd, making it necessary to  
761 either retrain the algorithms to include new individuals, or use  
762 an approach that is more suitable for an open herd setting, such  
763 as the one described by Andrew et al. (2021), that used deep  
764 metric learning to identify cattle that have never been seen be-  
765 fore by the network. However, this problem still needs to be ad-  
766 dressed in a larger scale in order to effectively implement visual  
767 identification systems in commercial farms or whole production  
768 systems, where hundreds or even thousands of individuals need  
769 to be identified and monitored simultaneously. Future applica-  
770 tions should be able to dynamically integrate new animals to  
771 the system as they are added to the herd, using mechanisms to  
772 tag images of never-before-seen animals for later identification.  
773 Potential approaches for addressing such problem are explored  
774 in the fields of self-supervised and zero-shot learning. Certain  
775 self-supervised learning techniques such as Contrastive Learn-  
776 ing allow neural networks to extract semantic representations  
777 from high-dimensional data using unlabeled datasets (Le-Khac  
778 et al., 2020), which could then be used to identify and clus-  
779 ter new examples of individuals that had not been seen before  
780 by the system, creating a temporary label that could later be  
781 mapped to a cow identification number. Furthermore, zero-shot  
782 learning consists of classifying samples that belong to classes not

783 observed during training, given some auxiliary information, and  
784 recently proposed methods have proven successful in areas of re-  
785 search regarding computer vision jointly with natural language  
786 processing (Xian et al., 2018).

#### 787 **4. Conclusion**

788 The outcomes of this study show that it is possible to use  
789 computer vision systems to identify individual animals using the  
790 3D surface of their dorsal body region. Both 2D and 3D rep-  
791 resentations of the dorsal surface, and the corresponding neural  
792 network architectures, can be used in such systems, each being  
793 more appropriate for different scenarios. Additionally, the ex-  
794 periments using images of calves taken during a period of intense  
795 growth provided evidence that neural networks can learn unique  
796 biometric features from the back of these animals, which remain  
797 recognizable even as body size changes. These findings suggest  
798 that it is possible to use neural networks to monitor and iden-  
799 tify animals from an early stage of life, or as they experience  
800 rapid body changes. By using exclusively the body shape of  
801 the animals (either through 2D depth images or 3D voxels and  
802 point clouds), the proposed methods may potentially be applied  
803 to species and breeds from which individuals share similar coat  
804 color patterns, which would be impossible to recognize using  
805 RGB images. This contributes to a broader application of ani-  
806 mal traceability and integrated phenotyping based on computer  
807 vision, facilitating infectious disease control, and improving farm  
808 productivity, food safety, consumer trust, and production sus-  
809 tainability. Future research can be developed towards investi-  
810 gating techniques such as semi-supervised and active learning,  
811 as well as hybrid approaches that merge traditional computer

812 vision methods and deep learning, to provide systems that are  
813 more data efficient and potentially perform better when exposed  
814 to large amounts of unlabeled data. Additionally, future work  
815 pertaining to computer vision-based identification systems in  
816 large commercial farms should evaluate the potential of novel  
817 self-supervised, zero-shot learning, and other techniques to over-  
818 come the challenge concerning dynamically changing herds.

## 819 **5. Acknowledgments**

820 This research was performed using the computational re-  
821 sources and assistance of the University of Wisconsin-Madison  
822 Center for High Throughput Computing (CHTC) in the Depart-  
823 ment of Computer Sciences. The CHTC is supported by Uni-  
824 versity of Wisconsin-Madison, the Advanced Computing Initia-  
825 tive, the Wisconsin Alumni Research Foundation, the Wisconsin  
826 Institutes for Discovery, and the National Science Foundation,  
827 and is an active member of the Open Science Grid, which is sup-  
828 ported by the National Science Foundation and the U.S. Depart-  
829 ment of Energy’s Office of Science. The authors would like to  
830 thank the financial support from the USDA National Institute  
831 of Food and Agriculture (Washington, DC; grant 2020-67015-  
832 30831) and Hatch project (WIS03085).

## 833 **References**

834 Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z.,  
835 Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M.,  
836 Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M.,  
837 Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg,  
838 J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C.,  
839 Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar,

- 840 K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F.,  
841 Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu,  
842 Y., and Zheng, X. (2015). TensorFlow: Large-scale machine  
843 learning on heterogeneous systems. Software available from  
844 tensorflow.org.
- 845 Aijazi, A. K., Checchin, P., and Trassoudaine, L. (2013). Seg-  
846 mentation based classification of 3d urban point clouds: A  
847 super-voxel based approach with evaluation. *Remote Sensing*,  
848 5(4):1624–1650.
- 849 Andrew, W., Gao, J., Mullan, S., Campbell, N., Dowsey, A. W.,  
850 and Burghardt, T. (2021). Visual identification of individual  
851 holstein-friesian cattle via deep metric learning. *Computers*  
852 *and Electronics in Agriculture*, 185:106133.
- 853 Andrew, W., Greatwood, C., and Burghardt, T. (2017). Visual  
854 localisation and individual identification of holstein friesian  
855 cattle via deep learning. In *Proceedings of the IEEE Inter-*  
856 *national Conference on Computer Vision Workshops*, pages  
857 2850–2859.
- 858 Andrew, W., Hannuna, S., Campbell, N., and Burghardt, T.  
859 (2016). Automatic individual holstein friesian cattle iden-  
860 tification via selective local coat pattern matching in rgb-d  
861 imagery. In *2016 IEEE International Conference on Image*  
862 *Processing (ICIP)*, pages 484–488. IEEE.
- 863 Awad, A. I. (2016). From classical methods to animal biomet-  
864 rics: A review on cattle identification and tracking. *Computers*  
865 *and Electronics in Agriculture*, 123:423–435.
- 866 Bello, R.-W., Talib, A. Z., Mohamed, A. S. A., Olubummo,

- 867 D. A., and Ootobo, F. N. (2020). Image-based individual cow  
868 recognition using body patterns. *Image*, 11(3).
- 869 Casino, F., Dasaklis, T. K., and Patsakis, C. (2019). A system-  
870 atic literature review of blockchain-based applications: Cur-  
871 rent status, classification and open issues. *Telematics and*  
872 *informatics*, 36:55–81.
- 873 Cheng, S., Leng, Z., Cubuk, E. D., Zoph, B., Bai, C., Ngiam,  
874 J., Song, Y., Caine, B., Vasudevan, V., Li, C., et al. (2020).  
875 Improving 3d object detection through progressive population  
876 based augmentation. In *European Conference on Computer*  
877 *Vision*, pages 279–294. Springer.
- 878 Cho, Y.-i. and Yoon, K.-J. (2014). An overview of calf diarrhea-  
879 infectious etiology, diagnosis, and intervention. *Journal of*  
880 *veterinary science*, 15(1):1–17.
- 881 Chollet, F. (2017). Xception: Deep learning with depthwise  
882 separable convolutions.
- 883 Chollet, F. et al. (2015). Keras. GitHub. Available at <https://github.com/fchollet/keras>.  
884
- 885 Cominotte, A., Fernandes, A., Dorea, J., Rosa, G., Ladeira, M.,  
886 van Cleef, E., Pereira, G., Baldassini, W., and Neto, O. M.  
887 (2020). Automated computer vision system to predict body  
888 weight and average daily gain in beef cattle during growing  
889 and finishing phases. *Livestock Science*, 232:103904.
- 890 Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-  
891 Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image  
892 Database. In *CVPR09*.

893 Dieleman, S., Schlüter, J., Raffel, C., Olson, E., Sønderby, S. K.,  
894 Nouri, D., Maturana, D., Thoma, M., Battenberg, E., Kelly,  
895 J., Fauw, J. D., Heilman, M., de Almeida, D. M., McFee, B.,  
896 Weideman, H., Takács, G., de Rivaz, P., Crall, J., Sanders,  
897 G., Rasul, K., Liu, C., French, G., and Degraeve, J. (2015).  
898 Lasagne: First release.

899 Dórea, J., French, E., and Armentano, L. (2017). Use of milk  
900 fatty acids to estimate plasma nonesterified fatty acid concen-  
901 trations as an indicator of animal energy balance. *Journal of*  
902 *Dairy Science*, 100(8):6164–6176.

903 Esslemont, R. and Kossaibati, M. (1999). The cost of respiratory  
904 diseases in dairy heifer calves. *The Bovine Practitioner*, pages  
905 174–178.

906 Fernandes, A. F., Dorea, J. R., and Rosa, G. J. (2020). Image  
907 analysis and computer vision applications in animal sciences:  
908 an overview. *Frontiers in Veterinary Science*, 7:800.

909 Gezawa, A. S., Zhang, Y., Wang, Q., and Yunqi, L. (2020). A re-  
910 view on deep learning approaches for 3d data representations  
911 in retrieval and classifications. *IEEE access*, 8:57566–57593.

912 Hahner, M., Dai, D., Liniger, A., and Van Gool, L. (2020).  
913 Quantifying data augmentation for lidar based 3d object de-  
914 tection. *arXiv preprint arXiv:2004.01643*.

915 Hansen, M. F., Smith, M. L., Smith, L. N., Salter, M. G., Bax-  
916 ter, E. M., Farish, M., and Grieve, B. (2018). Towards on-  
917 farm pig face recognition using convolutional neural networks.  
918 *Computers in Industry*, 98:145–152.

- 919 He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2018). Mask  
920 r-cnn.
- 921 Hinton, G., Srivastava, N., and Swersky, K. (2012). Neural  
922 networks for machine learning lecture 6a overview of mini-  
923 batch gradient descent. Available at [http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture\\_slides\\_lec6.pdf](http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf).  
924
- 925 Kaneene, J. B. and Hurd, H. S. (1990). The national animal  
926 health monitoring system in michigan. iii. cost estimates of  
927 selected dairy cattle diseases. *Preventive Veterinary Medicine*,  
928 8(2-3):127–140.
- 929 Kingma, D. P. and Ba, J. (2014). Adam: A method for stochas-  
930 tic optimization. *arXiv preprint arXiv:1412.6980*.
- 931 Le-Khac, P. H., Healy, G., and Smeaton, A. F. (2020). Con-  
932 trastive representation learning: A framework and review.  
933 *IEEE Access*, 8:193907–193934.
- 934 LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning.  
935 *Nature*, 521(7553):436–444.
- 936 Lin, M., Chen, Q., and Yan, S. (2014). Network in network.
- 937 Marcé, C., Guatteo, R., Bareille, N., and Fourichon, C. (2010).  
938 Dairy calf housing systems across europe and risk for calf in-  
939 fectious diseases. *Animal*, 4(9):1588–1596.
- 940 Maturana, D. and Scherer, S. (2015). VoxNet: A 3D Convolu-  
941 tional Neural Network for Real-Time Object Recognition. In  
942 *IROS*.
- 943 Nair, V. and Hinton, G. E. (2010). Rectified linear units improve  
944 restricted boltzmann machines. In *Proceedings of the 27th In-*

- 945 *ternational Conference on International Conference on Ma-*  
946 *chine Learning, ICML'10, page 807–814, Madison, WI, USA.*  
947 *Omnipress.*
- 948 Narayanan, P., Rander, P. W., and Kanade, T. (1998). Con-  
949 structing virtual worlds using dense stereo. In *Sixth Inter-*  
950 *national Conference on Computer Vision (IEEE Cat. No.*  
951 *98CH36271)*, pages 3–10. IEEE.
- 952 Okura, F., Ikuma, S., Makihara, Y., Muramatsu, D., Nakada,  
953 K., and Yagi, Y. (2019). Rgb-d video-based individual identi-  
954 fication of dairy cows using gait and texture analyses. *Com-*  
955 *puters and Electronics in Agriculture*, 165:104944.
- 956 O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S.,  
957 Hernandez, G. V., Krpalkova, L., Riordan, D., and Walsh,  
958 J. (2019). Deep learning vs. traditional computer vision. In  
959 *Science and information conference*, pages 128–144. Springer.
- 960 Park, U., Tong, Y., and Jain, A. K. (2010). Age-invariant face  
961 recognition. *IEEE transactions on pattern analysis and ma-*  
962 *chine intelligence*, 32(5):947–954.
- 963 Perez, L. and Wang, J. (2017). The effectiveness of data aug-  
964 mentation in image classification using deep learning. *arXiv*  
965 *preprint arXiv:1712.04621*.
- 966 Pini, S., Borghi, G., Vezzani, R., Maltoni, D., and Cucchiara,  
967 R. (2021). A systematic comparison of depth map represen-  
968 tations for face recognition. *Sensors*, 21(3):944.
- 969 Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2016). Pointnet:  
970 Deep learning on point sets for 3d classification and segmen-  
971 tation. *arXiv preprint arXiv:1612.00593*.

- 972 Qian, N. (1999). On the momentum term in gradient descent  
973 learning algorithms. *Neural Networks*, 12(1):145–151.
- 974 Robbins, H. and Monro, S. (1951). A stochastic approximation  
975 method. *The Annals of Mathematical Statistics*, pages 400–  
976 407.
- 977 Rousseeuw, P. J. and Croux, C. (1993). Alternatives to the me-  
978 dian absolute deviation. *Journal of the American Statistical*  
979 *association*, 88(424):1273–1283.
- 980 Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., and Szeliski,  
981 R. (2006). A comparison and evaluation of multi-view stereo  
982 reconstruction algorithms. In *2006 IEEE computer soci-*  
983 *ety conference on Computer Vision and Pattern Recognition*  
984 *(CVPR'06)*, volume 1, pages 519–528. IEEE.
- 985 Settles, B. (2009). Active learning literature survey.
- 986 Simonyan, K. and Zisserman, A. (2014). Very deep convolutional  
987 networks for large-scale image recognition. *arXiv preprint*  
988 *arXiv:1409.1556*.
- 989 Soilán Rodríguez, M., Lindenbergh, R., Riveiro Rodríguez, B.,  
990 Sánchez Rodríguez, A., et al. (2019). Pointnet for the au-  
991 tomatic classification of aerial point clouds. *ISPRS Annals*  
992 *of Photogrammetry Remote Sensing and Spatial Information*  
993 *Sciences*, pages 445–452.
- 994 Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov,  
995 D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Go-  
996 ing deeper with convolutions. In *Proceedings of the IEEE Con-*  
997 *ference on Computer Vision and Pattern Recognition*, pages  
998 1–9.

- 999 Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z.  
1000 (2016). Rethinking the inception architecture for computer  
1001 vision. In *Proceedings of the IEEE Conference on Computer  
1002 Vision and Pattern Recognition*, pages 2818–2826.
- 1003 Theano Development Team (2016). Theano: A Python frame-  
1004 work for fast computation of mathematical expressions. *arXiv  
1005 e-prints*, abs/1605.02688.
- 1006 Voulodimos, A., Doulamis, N., Doulamis, A., and Protopa-  
1007 padakis, E. (2018). Deep learning for computer vision: A brief  
1008 review. *Computational intelligence and neuroscience*, 2018.
- 1009 Voulodimos, A. S., Patrikakis, C. Z., Sideridis, A. B., Ntafis,  
1010 V. A., and Xylouri, E. M. (2010). A complete farm manage-  
1011 ment system based on animal identification using rfid technol-  
1012 ogy. *Computers and Electronics in Agriculture*, 70(2):380–388.
- 1013 Wang, P.-S., Liu, Y., Guo, Y.-X., Sun, C.-Y., and Tong, X.  
1014 (2017). O-cnn: Octree-based convolutional neural networks  
1015 for 3d shape analysis. *ACM Transactions on Graphics (TOG)*,  
1016 36(4):1–11.
- 1017 Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey  
1018 of transfer learning. *Journal of Big Data*, 3(1):9.
- 1019 Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and  
1020 Xiao, J. (2015). 3d shapenets: A deep representation for vol-  
1021 umetric shapes. In *Proceedings of the IEEE conference on  
1022 Computer Vision and Pattern Recognition*, pages 1912–1920.
- 1023 Xian, Y., Lampert, C. H., Schiele, B., and Akata, Z. (2018).  
1024 Zero-shot learning—a comprehensive evaluation of the good,

- 1025 the bad and the ugly. *IEEE transactions on pattern analysis*  
1026 *and machine intelligence*, 41(9):2251–2265.
- 1027 Yao, L., Hu, Z., Liu, C., Liu, H., Kuang, Y., and Gao, Y. (2019).  
1028 Cow face detection and recognition based on automatic fea-  
1029 ture extraction algorithm. In *Proceedings of the ACM Turing*  
1030 *Celebration Conference-China*, pages 1–5.
- 1031 Yukun, S., Pengju, H., Yujie, W., Ziqi, C., Yang, L., Baisheng,  
1032 D., Runze, L., and Yonggen, Z. (2019). Automatic monitor-  
1033 ing system for individual dairy cows based on a deep learning  
1034 framework that provides identification via body parts and es-  
1035 timation of body condition score. *Journal of Dairy Science*,  
1036 102(11):10140–10151.
- 1037 Zhu, X. J. (2005). Semi-supervised learning literature survey.