Exploring a Siamese Neural Network as a novel individual cow identification technique for daily monitoring free water intake

Daniël M. van Herwijnen [1] and Miel Hostens [*]

[1] *First author: Daniël M. van Herwijnen*
[*] *Corresponding author: Miel Hostens, m.m.hostens@uu.nl*

Abstract

Water is an essential nutrient for a healthy and high performing dairy herd, since most of all life's processes require water. Particularly, as a cow produces more milk, its water requirement increases. For this reason, insight into the drinking behaviour of a dairy herd is important, but the currently available data is outdated. Furthermore, daily free water intake of individual cows is currently not being monitored, even though the importance of water is recognized. Consequently, it is not possible to make sure all cows' water requirements are met. There is a strong need for an efficient technique for monitoring free water intake for individual cows. Computer vision could be used to carry out almost all tasks required for such a monitoring task. For monitoring, detecting drinking behaviour and individual cow identification are required. We explored an approach for individual cow identification through training a convolutional neural network with a triplet loss, called a siamese neural network. The siamese neural network takes an image as input and outputs an n-dimensional feature vector, also called an embedding. The triplet loss encourages small distances between embeddings of the same cow and large distances between embeddings of different cows. Ultimately, the neural network can learn to differentiate between cows' spot patterns rather than directly learning the identifiers corresponding to the images. After training, the model should be able to distinguish cows' spot patterns well enough that it only needs one or a few images per cow to correctly classify all cows in the herd. For this reason, the classification is called one-shot classification. During inference, an embedding is calculated for a new image and a prediction is based on the most similar embedding in a database. Our models appeared to perform well, but upon further inspection seemed to be confounded by the image backgrounds. When removing the background, accuracy dropped radically. We propose this can be addressed by using instance segmentation to remove the background from the training images. The model seems promising but needs to be developed further in order to use it in individual cow identification. Once this is achieved, our model could be combined with a pose estimation model like DeepLabCut. Using pose estimation, it is possible to determine where a cow is located and how it is oriented. This can be used to predict whether a cow is drinking from a water trough. Altogether, this can be utilized to create a technique for monitoring the time a cow spends drinking. Combined with a water flow meter, an individual free water intake monitor can be realized for monitoring dairy herds. This will give us insight in whether the individual water requirements of cows can be met in the current dairy farms.

1 Layman's Summary

A dairy cow needs enough water to produce good quality milk. Over the last decades, the amount of milk produced per cow has increased a lot. This means that the amount of water needed per cow has increased as well. It is unclear whether cows nowadays can drink enough water at the dairy farm, because individual water intake is currently not being monitored. An easy way of measuring the amount of water a cow drinks needs to be developed and artifical intelligence can be used to achieve just that. Using one camera per drinking trough, we want to survey the drinking behaviour. A cow can be detected in an image using algorithms like YOLOv5. We then need to detect whether it is drinking. By locating its head using an algorithm called DeepLabCut, we can predict a cow is drinking when its head is close to the drinking trough. Next, the cow needs to be identified in order to track its water consumption. We developed a model with the aim of doing this, called a siamese neural network. It can learn to 1) distinguish images of different cows and at the same time 2) learn to recognize different images of the same cow. These steps combined can let us analyze a video and track when a cow was drinking and which cow it was. The siamese neural network seems to be promising but needs to be improved by removing the background from an image, such that only the cow is visible. When this is achieved, we can easily measure the time a cow spends drinking. Combined with data from water flow meters, we can monitor how much water a cow drinks.

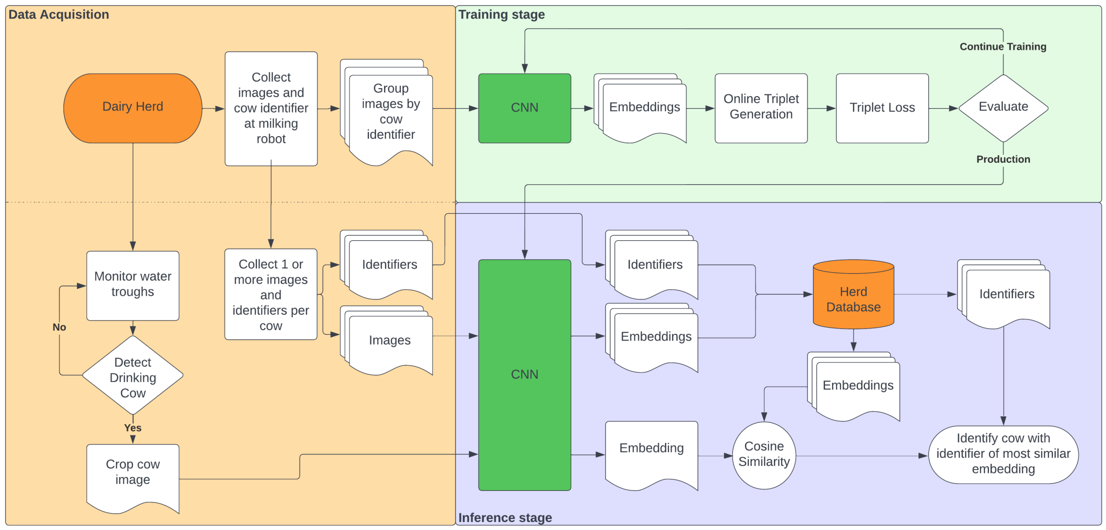

Figure 1. Proposed workflow of the drinking monitor. To train the CNN, images for every cow in a herd are collected and grouped by identifier. Online triplet generation is performed within mini-batches and the loss is calculated. In the inference stage there is no more need for triplets. For each cow in a herd, one or more images are collected and and embedding is calculated for each image. The embedding is stored together with the corresponding identifier. During monitoring, when a drinking cow is detected (for example using DeepLabCut), the cow is cropped from the frame and an embedding is calculated. Then this embedding is compared to all embeddings in the herd database and the most similar embedding will predict the corresponding identity of the drinking cow.

2 Introduction

2.1 Water Consumption

The most important nutrient for a healthy and high performing dairy herd is considered to be water (Houpt 1984). It is required for almost all life's processes (e.g. nutrient digestion and metabolism, body heat regulation, maintaining fluid/ion-balance and waste excretion) (Houpt 1984; Murphy 1992). Losing 20 percent of the body water is fatal. (Houpt 1984). In adult cattle, water deprivation leads to a reduction in milk production (Little *et al.* 1980) and depriving calves of water leads to limited weight gain (Kertz, Reutzel, and Mahoney 1984). It is sufficiently clear that livestock need enough water. However, over the last decades, concerns have been raised about the sustainability of dairy industries due to climate change, water-shortage (Steinfeld *et al.* 2006; Ridoutt *et al.* 2010; Cardot, Roux, and Jurjanz 2008) and the health and welfare of dairy cows (Barkema *et al.* 2015); Milk yield per cow has strongly increased over the last decades (Zehetmeier *et al.* 2012), though it is unclear whether cows are also allowed to drink more water. Therefor, it is important to monitor the water consumption of dairy cows, especially considering that the available data is no longer proportionate to the current milk yield per cow. (Houpt 1984; Murphy 1992; Jensen and Vestergaard 2021).

2.2 Monitoring Free Water Intake

Through technical advancements of the last decades, monitoring and detection has been progressively automated. Moreover, collecting data that used to be infeasible when performed manually is now possible due to automation, like tracking water consumption of dairy cows (Cardot, Roux, and Jurjanz 2008; Meyer *et al.* 2004). Despite the importance of water consumption for animal welfare, methods for daily monitoring individual Free Water Intake (FWI) are lacking (Cardot, Roux, and Jurjanz 2008). FWI is particularly measured when experimentally required, but not on a daily basis for monitoring. During such experiments, FWI is measured through automatic water flow registration (Melin, Wiktorsson, and Norell 2005) or still by manually reading water meters of individual stalls (Huuskonen, Tuomisto, and Kauppinen 2011; Liang, Hudson, and Ballou 2020). Having a method for automatic FWI monitoring should be a priority to ensure all cows can drink enough water.

2.3 Identification Problem

An important aspect in monitoring is identification. Radio-Frequency Identification (RFID) is a well known and widely used way of identifying animals (Mendes *et al.* 2011; Matthews *et al.* 2016). Despite being reliable and widely used, RFID has major drawbacks. First, RFID is an invasive method of data collection, since an animal must be outfitted physically with an RFID-tag. Secondly, because each individual requires an RFID-tag, the total cost scales with the number of animals. Finally, detection relies on proximity, thus every location where data collection takes place has to be equipped with an RFID-scanner. We propose using a computer vision as novel identification technique for cows.

2.4 Intro to using vision

Computer vision is an interdisciplinairy field in science that aims to model the human visual system such that computers can perform some tasks like humans (Huang 1996). Computer vision can be subdivided into several sub-tasks, among which are object detection, object recognition and pose estimation. Yolov5 is an example of object detection and recognition (Jocher *et al.* 2020). Trained on the COCO-dataset (Lin *et al.* 2014), Yolov5 can detect and recognize 80 different objects, including cows (Jocher *et al.* 2020). The Yolo-architecture is open source and can easily be trained on a custom dataset. DeepLabCut (DLC) is an example of pose estimation (Mathis *et al.* 2018). Using their graphical user-interface (Nath *et al.* 2019), data can be easily labelled and used to train their model. Using these methods, an individual drinking monitor could be realized based on computer vision. DLC can be used to determine the position of a cows' body parts. Using these positions, it could be determined whether a cow is drinking. For example, when the mouth of a cow is close to the water trough, it is more likely to be drinking than when facing away from it. To monitor the drinking behaviour, identification of drinking cows needs to be addressed, for which we propose a Siamese Neural Network (SNN).

2.5 Siamese Neural Network

A SNN is an example of similarity learning. We defined a training stage and an inference stage, illustrated in figure 1. In the training stage, instead of learning directly which label corresponds to which image, a SNN learns to distinguish between different objects without predicting the correct label yet. In comparison, even without knowing what an elephant and a monkey are, a human can visually distinguish these two animals quickly. Only after learning to distinguish individual cows, the model will use the corresponding labels in the inference stage to classify new images. In the training stage, more images per cow and images in total can speed up the learning process. Once the inference stage is reached, one image -or a few images- per cow should be sufficient to be able to classify individual cows; after all, the SNN has learned to distinguish individual cows. In comparison, giving a human that has learned to distinguish an elephant from a monkey what the name of each animal is, it should be able to classify an elephant and a monkey. Hence, in the inference stage, a SNN is a one-shot classifier, because it only needs one or a few example images to classify a cow correctly (Koch, Zemel, and Salakhutdinov 2015; Schroff and Philbin 2015).

A SNN is a model that converts images through convolution into n-dimensional feature vectors, also called embeddings. It then learns in the training stage to make embeddings of the same cow more similar and embeddings of different cows less similar. Having reached the inference stage, classification is simply comparing the embedding from a new cow image to the embeddings of cow images in a database; the most similar is then most likely the same cow (Koch, Zemel, and Salakhutdinov 2015; Schroff and Philbin 2015).

An important benefit of a SNN is that the need for retraining is much lower when adding a new cow to the herd. The model does not directly learn which spot pattern belongs to which individual but rather learns how to distinguish different spot patterns. Assuming the model can generalize well, only one image of a new cow is needed to update the database without any retraining. To achieve generalization, training on a large enough dataset that captures all possible variation in cows' spot patterns is required (Wang *et al.* 2014; Schroff and Philbin 2015). Consequently, an SNN trained on herd A is not guaranteed to be able to distinguish all cows in herd B, but as the number of cows in herd A increases, the model should be able to get better at distinguishing cows in herd B.

Triplet Loss

A SNN is often depicted like a network with multiple identical Convolutional Neural Network (CNN)s stacked on top of each other. This visual aid can be misleading since there is only one CNN. Comparing the outputs of this CNN is achieved through a triplet loss. A triplet loss groups three embeddings into triplets: an anchor (x_a), a positive (x_p) and a negative (x_n) where x_a and x_p are from the same cow but x_a and x_n are from different cows. It then penalizes a) large distances (D) between embeddings of the same cow $D(x_a, x_p)$ and b) small distances between embeddings of different cows $D(x_a, x_n)$. The grouping into triplets is the motivation behind the stacked visualization.

Offline Triplet Generation

There are two main ways of implementing a triplet loss; offline and online (Schroff and Philbin 2015). Offline triplet generation occurs before training starts or between training epochs. The data is organized into triplets (x_a,x_p,x_n), the triplets are split into batches and then training starts on these batches. Despite benefiting from very short computation time, the performance can be very poor (Schroff and Philbin 2015). Because the weights are updated such that $D(x_a, x_p)$ becomes small and $D(x_a, x_n)$ becomes large, offline-generated triplets will lose information for the network to learn over time trained. This can be circumnavigated by regenerating the triplets every i epochs (Schroff and Philbin 2015).

Online Triplet Generation

Instead of making batches of triplets, online triplet generation is used within mini-batches to find the most informative triplets after they are converted into embeddings. Assuming each embedding can be an anchor x_a, the positive x_p with the largest $D(x_a, x_p)$ has the most impact on training. Conversely, the negative x_n that has the most impact on training has the lowest $D(x_a, x_n)$ (Schroff and Philbin 2015). Within each batch, the furthest x_p and closest x_n are calculated for each x_a. This takes more time, but generally yields higher performance gain (Schroff and Philbin 2015).

3 Materials and methods

3.1 Used Data Sets

Drinking monitor

Six camera's were installed on a dairy farm in the Netherlands at six different drinking troughs. 24 hours of video footage was collected on October 10 2022.

LR-120

On the same farm, four camera's were installed at the automated milking robots. The cameras were mounted at the entrance of the robot pointing down to see the top side of each cow (figure 2). 221 cows

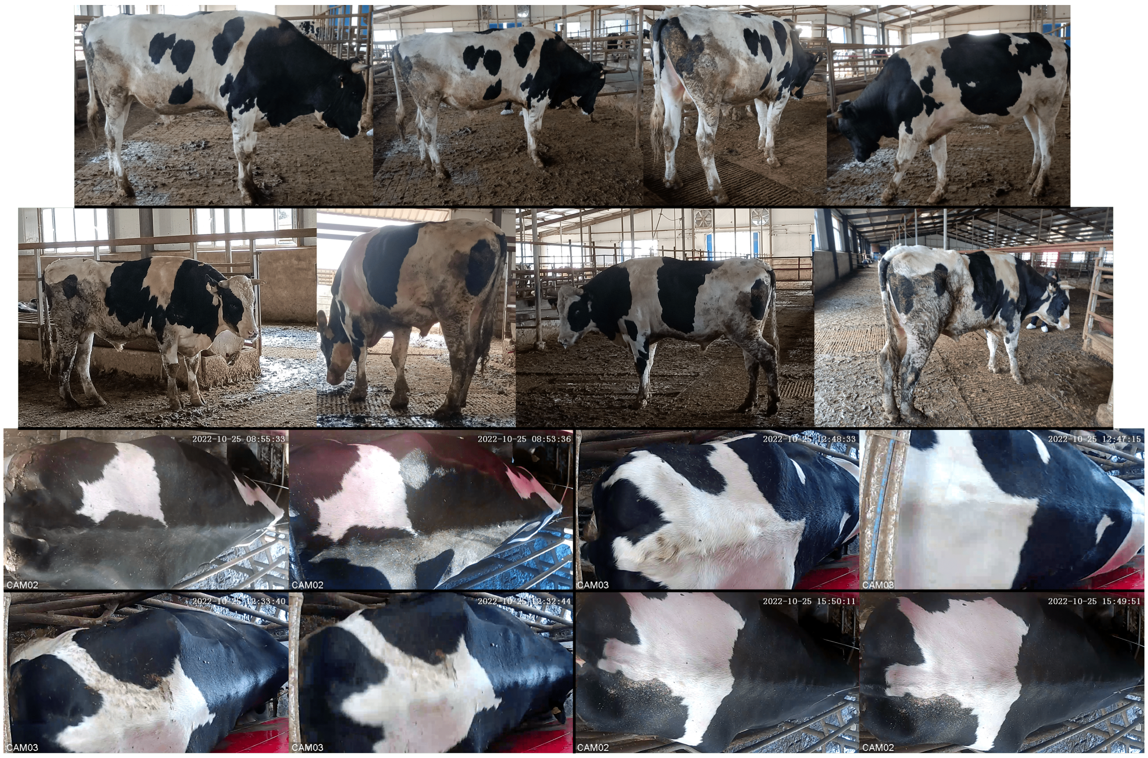

Figure 2. Overview of some images from LR-120 and IC-64. The top two rows display two cows from IC-64. The top row shows high background and pose similarity between different images of the same cow. The second row shows more variety. The two bottom rows show four cows from LR-120. Images from different cows are separated by a black line.

had been recorded. During one milking session of a cow, twenty images were collected manually from the video footage for a 120 cows. Only 120 cows were used to verify the model before using all data, because extraction is a time consuming process. The images were collected from 3 different milking robots. To summarize, the dataset contains 2400 images of 120 different cows and is split 80%/20% into train and validation. The dataset is referred to using the abbreviation LR-120.

All frames in the videos had a timestamp and a cam-label. These were blurred in all images to prevent the model from learning the time at which any cow was at the milking robot or which camera shot the footage (see figure 3).

IC-64

A dataset containing 1485 images of 13 different cows was published by Li *et al.* 2021. It contains cows from different angles with different lighting and background (figure 2). A subset of this dataset has been used. It is composed of 80 images per cow for a total of 1040 and is split 80%/20% into train and validation. It is referred to as IC-64. This dataset was used because of it's high number of images per cow, low number of cows and good image quality. It was used to test whether the quality of our dataset impacted the performance of the model. We only used a subset of the dataset because we needed an equal number of images per class for our implementation.

3.2 Model Architecture

The model architecture can be split into two stages; features extraction and feature mapping. The first stage extracts features from the input image using multiple convolutions. The second stage converts the features into a 128-dimensional vector. The main differences between our models comprise changes in these stages. The second stage has not been changed much during the project. Most of the experiments have been conducted on the feature extraction stage.

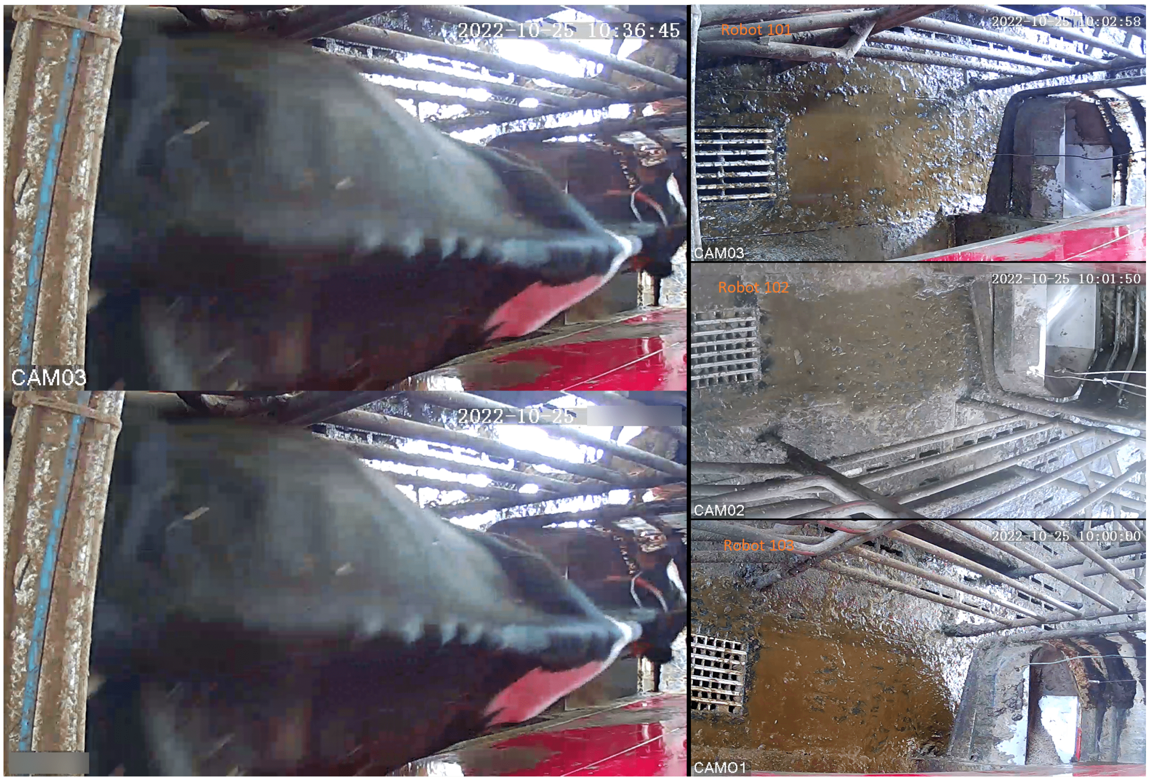

Figure 3. Image blurring and robot backgrounds. The top left shows a frame from the recordings. In the bottom left image, the cam-label and timestamp have been blurred. The three images on the right show the image backgrounds for robot 101, 102 and 103. There are clear features by which to recognize which camera took the picture.

SNNv1

SNNv1 is based on ResNet50, which is composed of 5 stages of multiple convolutional blocks. The pre-trained ImageNet weights for ResNet50 are loaded and all layers up untill conv5-block1-out are set to untrainable; all following layers are trainable. ResNet50 is followed by global average pooling and a single fully connected layer which outputs a 128-dimensional vector. It has been trained using a custom online triplet loss. A random contrast-augmentation-layer was used to make the model generalize better to unseen data.

SNNv2

SNNv2 is not based on already existing models but is built from scratch. After random contrast-, brightness- and rotation-layers, the input is downsampled by a convolution block which consist of a convolution, layer normalization and leaky-relu activation. After the downsampling, a series of basic-blocks is used that are defined as a max pooling, followed by three convolution blocks. A skip connection is added between the output of the first convolution block and the output of the third convolution block before the leaky-relu activation. After 5 basic blocks, there are two 1028-neuron fully connected layers and one fully connected layer that outputs a 128-dimensional vector (see figure 4 for an overview of the model).

SNNv3

SNNv3 is based on Google's InceptionResnet (Szegedy *et al.* 2016). Pretrained weights are not used but the weights are randomly initialized instead. The inception network is followed by two 1028-neuron fully connected layers and one fully connected layer that outputs a 128-dimensional vector. The input data is augmented through random brightness- and contrast- and translation-layers. The translation layer shifts the image horizontally and vertically to make the model less reliant on the exact position of the cow.

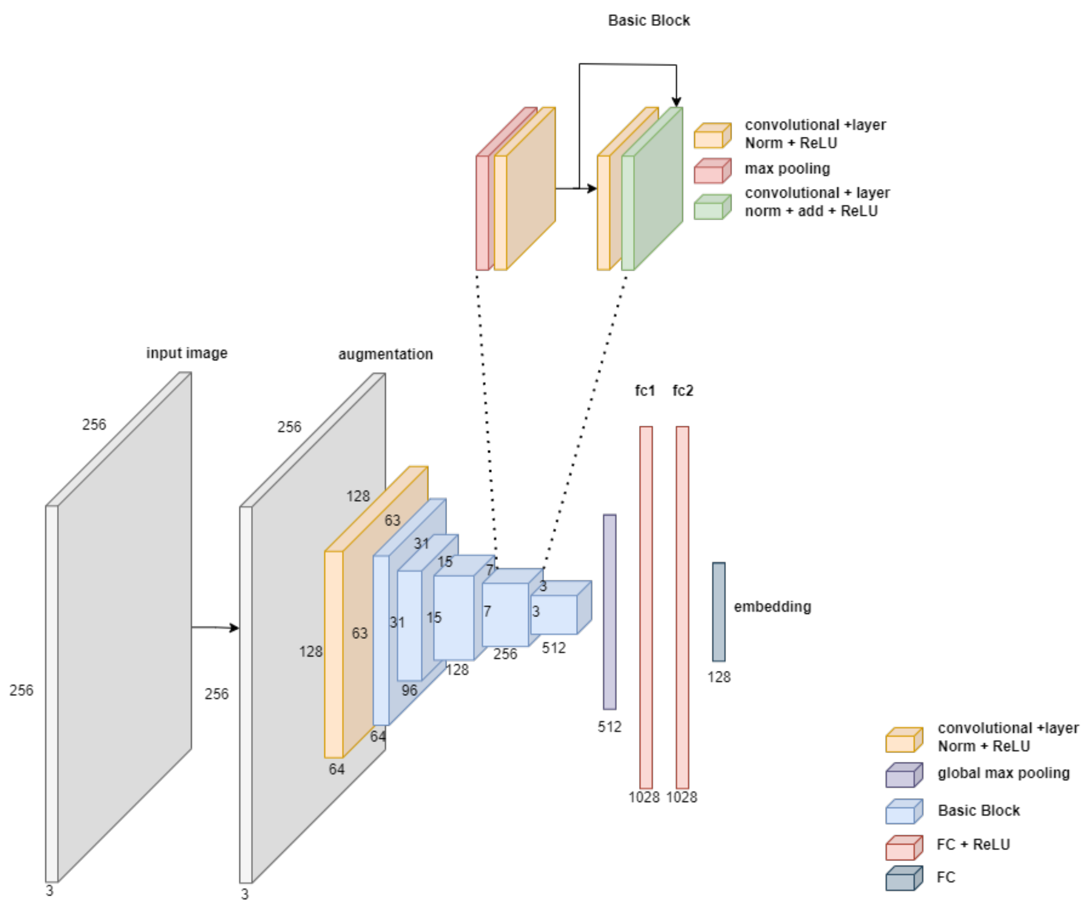

Figure 4. SNNv2 model architecture. The model is comprised of a series of basic blocks. The input size is 256x256. After feature extraction in the convolutional layers, a global max pooling converts the outputs of the last convolution to a 1-dimensional vector. After three fully connected layers, the features are mapped to a 128-dimensional embedding.

3.3 Silhouette Score

A metric was needed to evaluate the training progress aside from the loss. Since the model is learning how to make embeddings that are similar for the same cow and less similar for different cows, we used the silhouette score as a measure of how well the model is clustering its embeddings. The silhouette score describes how well any one embedding *i* has been clustered with respect to all other embeddings. First, the average distance from *i* to all embeddings with the same label are calculated ($a(i)$). Next, the average distance from *i* to all embeddings of the closest cluster are calculated ($b(i)$). The silhouette score can then be calculated using

$$s(i) := \frac{b(i) - a(i)}{max(a(i), b(i))}$$

The silhouette score ranges from -1 (worst clustering possible) to 1 (perfect clustering). Calculating the silhouette score at each training step helps us track the training progress (Rousseeuw 1987).

3.4 Cosine Similarity

The cosine similarity describes the similarity between two vectors A and B (Han, Kamber, and Pei 2012) and is defined as

$$\cos(\theta) := \frac{A \cdot B}{||A|| \times ||B||}$$

Cosine similarity ranges from -1 to 1 where 1 means the vectors are equal and -1 means the vectors are opposites. Given a set of embeddings of which we know the corresponding label, we can predict the label of an unknown embedding by taking the embedding with the highest cosine similarity as the unknown embedding. The cosine similarity is used in the inference phase of our SNN. An embedding is calculated for an image and the most similar reference embedding will predict the identify of the cow in the image.

3.5 Python Code and libraries

All code was written in Python 3.8.13. The neural network where made with TensorFlow 2.10.0 and the triplet loss from TensorFlow Addons 0.18.0 was used. All code is made available through GitHub and all dependencies can be found in requirements.txt. Most figures in this paper were also code generated. The code to replicate them can be found in the figures folder on github.

4 Results

4.1 Triplet Loss

The earliest versions of the SNN all had offline triplet generation where triplets were only generated at the beginning of training. As mentioned before, this is very inefficient and the models did not perform well. After finding the paper about FaceNet (Schroff and Philbin 2015) a lot of improvements were made which resulted in a first version of the model, SNNv1.

4.2 Architecture Experiments

A next step to update the model was by running a number of experiments changing different parts of the model. The model was trained with multiple different configurations for 50 epochs.

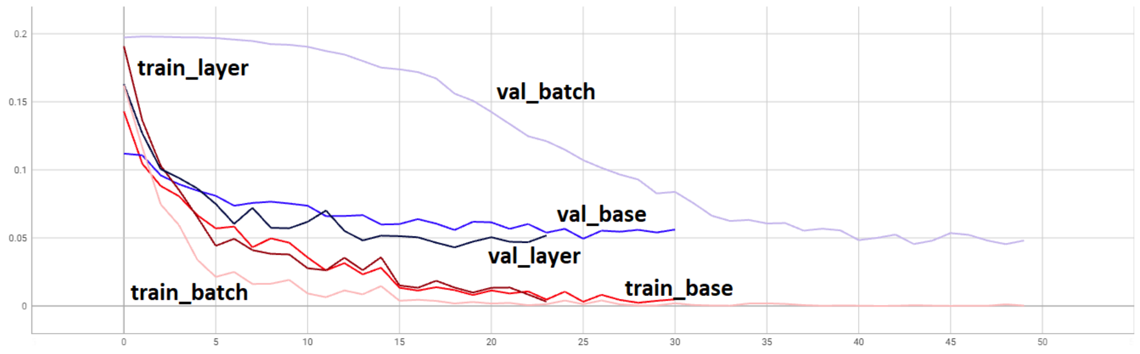

Figure 5. Train and validation loss for different normalizations. Three models were trained: one without normalization, one with layer normalization and one with batch normalization.

Layer normalization

Normalizing the activations inside the network is an important step for deep CNNs since it ensures proper gradient back-propagation and reduces training time (Ba, Kiros, and Hinton 2016). There are different methods of normalizing in a deep learning. To know which normalization would work best for our model, one model was trained without normalization and two with layer normalization or batch normalization. Figure 5 shows that there is little difference in training loss for all three models. However, there is a difference for the validation loss. Both layer and batch normalization converge towards 0.05, but layer normalization is much closer to the base model. The different graph-lengths are caused by early stopping.

Random Contrast

Figure 6 shows the silhouette score for different values of random contrast augmentation. Without random contrast there was a sizable difference between the train and validation silhouette score. Random contrast

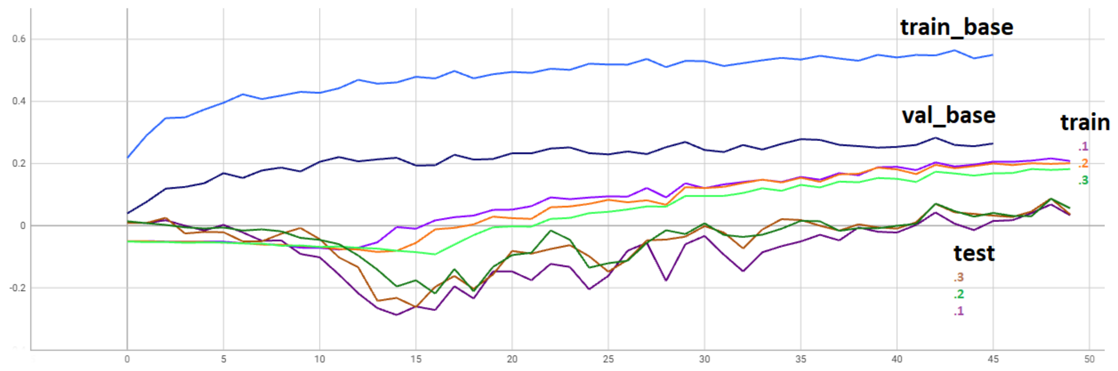

Figure 6. Train and validation silhouette score for different values of random contrast. Four models were trained: one without random contrast, and three with a random contrast of 0.1, 0.2 or 0.3.

appeared to have a negative correlation with the training silhouette score; higher levels of random contrast led to slower converging. The difference between validation and training scores became smaller when random contrast was increased. 0.1 random contrast had a consistently lower validation silhouette score. 0.2 and 0.3 were very comparable. The same experiment has been conducted for random brightness. However, due to very quick early stopping, the experiment was inconclusive.

4.3 Embedding Classification

Three different methods were used to predict the labels based on the produced embeddings: K-Nearest Neighbors classification (KNN), Support Vector Machine (SVM) and cosine similarity. Accuracy was calculated for each for three different models. The first two models were SNNv2 trained on LR-120 and IC-64 respectively. The third model was SNNv3 and it was trained on LR-120. Accuracy was identical when switching cosine similarity. Figure 7 shows that the cosine similarity got the best accuracy on SNNv2. SVM gets the highest accuracy for SNNv3. The graphs appeared to follow the same trend amongst the three methods and the accuracy achieved with these methods was very similar.

4.4 Model Accuracy

All Data Classification

Three different models where trained to assess the performance of the SNN. SNNv2 was trained on LR-120 and IC-64 and SNNv3 was trained on LR-120. There was not enough time to train SNNv3 on IC-64. The validation embeddings were classified at five training epochs intervals by calculating their cosine similarity to the train embeddings. The label of the most similar embedding was the prediction. The predictions were compared to their respective validation labels and the accuracy was calculated (figure 8). Without much training, the accuracy was already above 0.70 and barely increased as training progressed.

One Shot Classification

As was mentioned before, after training, a SNN is a one-shot classifier. Instead of calculating the cosine similarity between a validation embedding and all training embeddings, for each cow two images (left and right side) were selected from IC-64. Using the IC-64 trained SNNv2 the $2 \times 13 = 26$ reference embeddings were used to predict labels for all validation embeddings and the accuracy was calculated. This way, the one-shot character is better captured since in theory all images of a single cow should result in a very similar embedding. As can be seen in figure 9 the accuracy after 5 epochs was much lower than when comparing to the entire training set. After 100 epochs, accuracy was only around 0.40. When the backgrounds of the 26 reference images was replaced by a blue background, the accuracy deteriorated (figure 9).

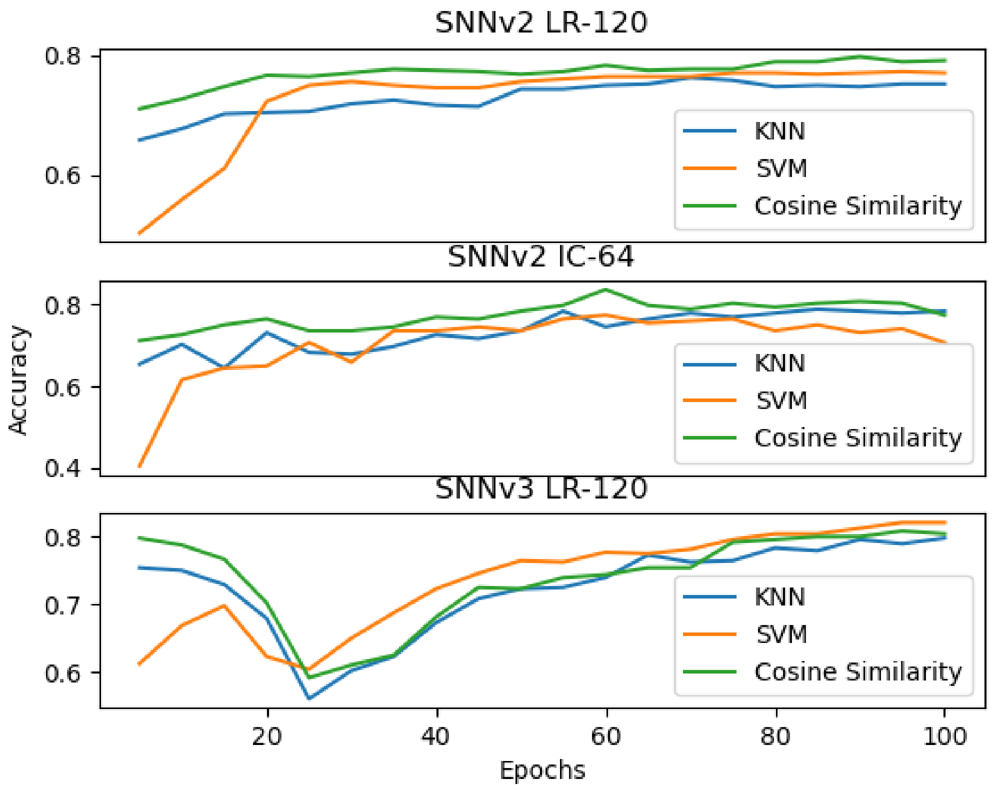

Figure 7. Validation accuracy when using KNN, SVM and cosine similarity. At 5-epoch intervals, the accuracy was calculated using three different methods: KNN, SVM and cosine similarity.

Different Robots
The influence of image backgrounds was also evaluated for LR-120. Due to the size of LR-120, background removal would be too time consuming. However, the images in LR-120 were collected from three different camera's at three different milking robots and the backgrounds from these images were distinguishable (figure 3). We analyzed how many of the misclassified images were classified to be a cow from a different robot. Figure 10 shows that an image is almost exclusively misclassified as a different cow from the same robot.

5 Discussion

5.1 Image Background

During experimentation, different models showed high classification accuracy (80%) which indicates good performance. However, most accuracy graphs show that the initial accuracy is already very high to begin with, implying an unidentified problem with out model. To identify the problem, dataset IC-64 was inspected, since the quality of this data is better and there are more images per cow. We observed strong similarity between images of the same cow. This can be explained by the fact that groups of images appeared to be taken within short succession. This means that a large number of images had the same background and the cow in those images showed very similar poses (figure 2). The images of each cow in LR-120 were collected during a single milking session per cow lasting a couple of minutes. We hypothesized that the background of an image had a very strong influence on the model's capacity to differentiate cows' images. This was confirmed by removing the background of the 26 IC-64 reference images (One Shot Classification), because accuracy dropped from 20-40% to 5-15%. Similarly, a model trained on

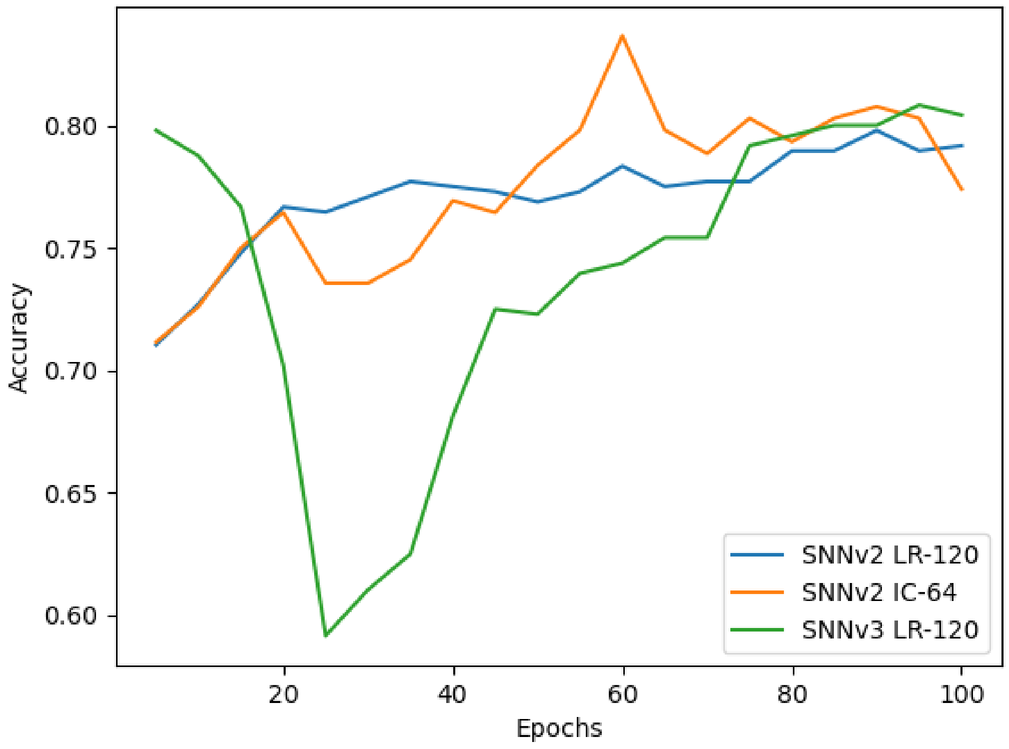

Figure 8. Validation accuracy every 5 training epochs. Using cosine similarity, the accuracy was calculated for the three different models.

LR-120 exclusively misclassified cows as cows that were recorded at the same milking robot (Different Robots), affirming the importance of the background to the model. This needs to be addressed in future development of our model.

Isolating a cow from an image and removing the background can be done using segmenation. In the beginning stages of the project, another dataset was used. All cow-objects were detected and cropped using YOLOv5 (Jocher *et al.* 2020) because some images contained multiple cows. Cropping resulted in segmentation. Importantly, the remaining backgrounds had very high variation because images were not collected at static locations. However, when switching to IC-64, the use of YOLOv5 was overlooked because all images contained one cow only and most of the pictures had relatively very little background. The same goes for LR-120, because there was only one cow at the milking robot and only a small portion of the background appeared to be visible. Even so, our results shows that the background still had a very large impact. This could not have been solved easily with YOLOv5, because only the cows' backs are visible, which makes detecting them difficult. Moreover, using YOLOv5 -or other object detection software- has the downside of having a bounding box. The cow-objects are not rectangular and therefor some of the background remains visible.

YOLOv5 belongs to the object localization tasks. Another computer vision task is segmentation, which predicts a label per pixel of an image. There are two main variants of segmentation: semantic and instance segmentation. Semantic segmentation aims to label which kind of object any pixel belongs to, whereas instance segmentation differentiates between different objects of the same class (Hafiz and Bhat 2020). I propose to use instance segmentation as a preprocessing step before training the SNN. With segmentation, the background of an image can easily be removed. Besides, if another cow is visible in the image, it can be removed when using instance segmentation as opposed to semantic segmentation.

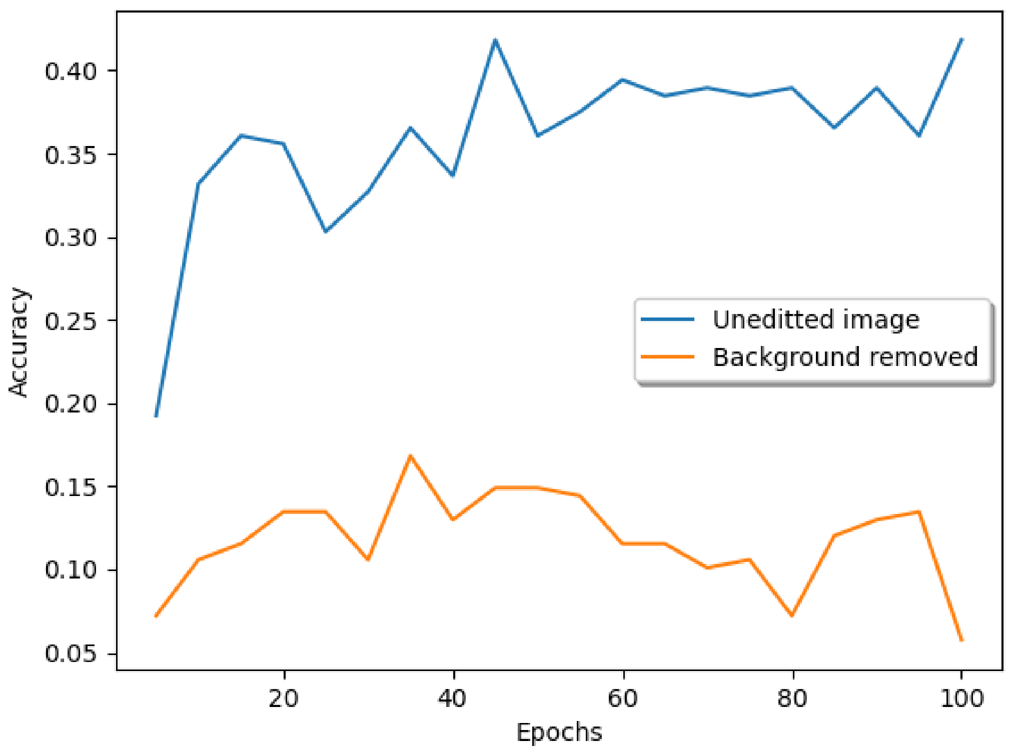

Figure 9. One-shot accuracy with and without background. Accuracy was calculated for SNNv2 trained on IC-64. To assess the generalization of the model, two images were selected to make reference embeddings. Next, the background of these images were removed.

Furthermore, instance segmentation can also be valuable when moving on to the monitoring of the drinking behaviour. When a frame is segmented into cow-objects, only cow-objects that are close to the drinking through have to be analyzed further. Moreover, instance segmentation might also be used to find the face and rear of a cow-object, eliminating the need for training a DLC-model for pose estimation altogether.

5.2 Architecture Experiments

Our most recent models all used layer normalization The validation loss converged equally fast or even faster with layer normalization than without normalization. Given a much faster convergence for layer normalization than for batch normalization, layer normalization appears to be better for our model. This is consistent with Ba, Kiros, and Hinton 2016. Increasing random contrast augmentation reduces overfitting. Our results (Random Contrast) show that increasing the random contrast decreases the difference between train and validation loss. This can be explained by the fact that random contrast increases the variation within the data. Too high levels of random contrast can however negatively impact training time substantially. Random contrast of 0.2 and 0.3 had a very similar validation loss. This indicates that increasing the random contrast from 0.2 to 0.3 has very little effect. Still, the training loss for 0.3 was lower that 0.2. We therefor chose a random contrast of 0.2 for our subsequent models. The architecture experiments were aimed to improve the accuracy because it appeared our model was starting to perform well. However, since the impact of image background was only identified after conducting the architecture experiments (Architecture Experiments), it is unclear whether these results can be carried over after implementing background removal. It would be advisable to revisit these experiments after updating the

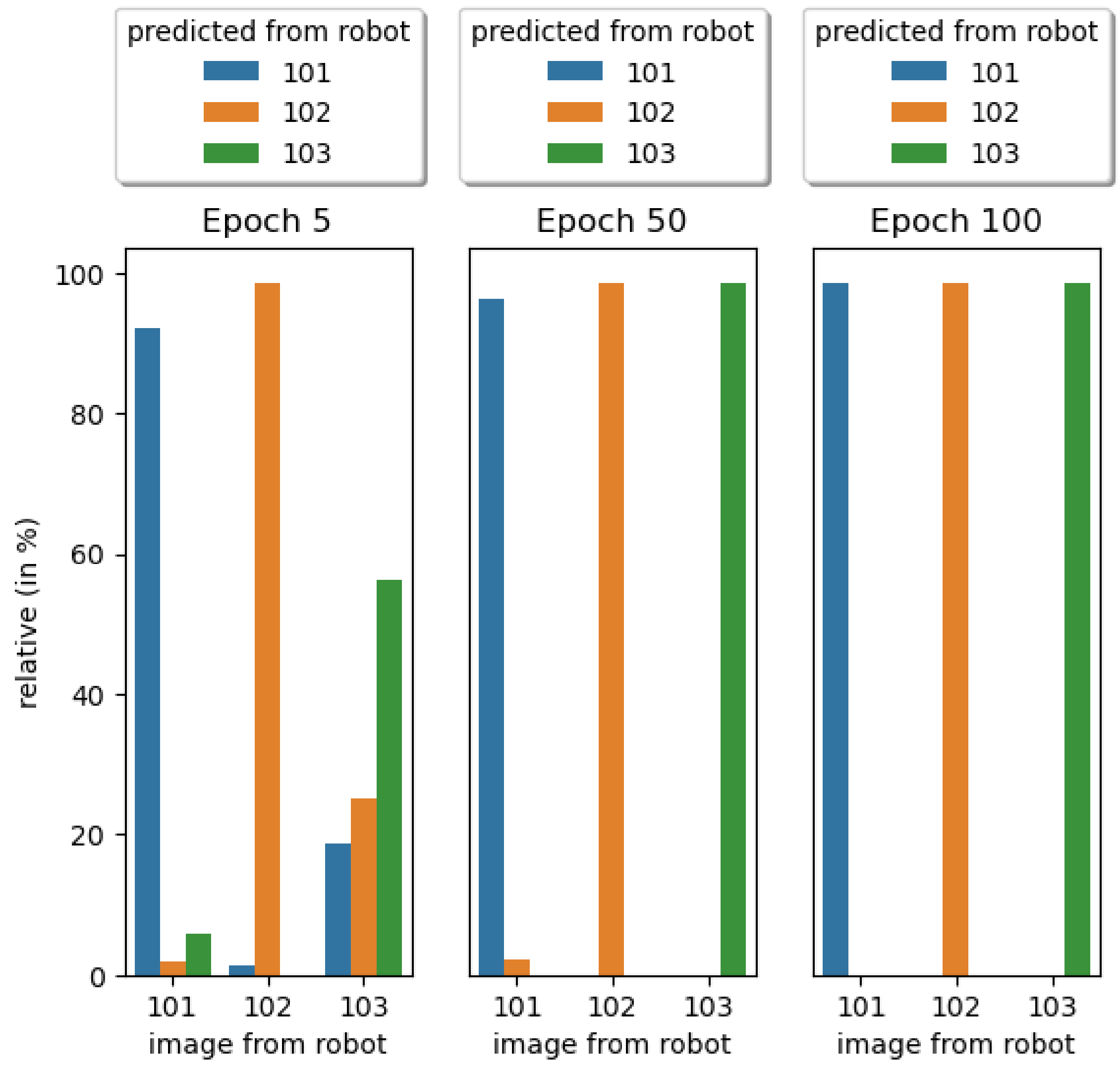

Figure 10. Percentage of misclassification between images taken from different robots. At 3 different epochs, only misclassifications were analyzed. An image was taken at one of three robots. A classification consists of an embedding that needs to be classified (target) and an embedding that is most similar to the target (prediction). For each misclassified target, the robot at which the prediction image was taken was compared to the target's robot.

model with instance segmentation.

5.3 Embedding Classification

During inference, the embedding of a new image has to be compared to the embeddings in the database. We compared three different methods for this comparison, namely KNN, SVM and cosine similarity. The accuracy for each of the three methods was very similar, except for the SVM-accuracy in the first few epochs. The high similarity between cosine similarity and KNN can be explained by the fact that both algorithms look for the nearest embedding. SVM on the other hand tries to fit a hyperplane to sepearte all classes. This can explain why SVM deviates more from KNN and cosine similarity. Even though the differences between the three methods appeared to be small, we still opted for cosine similarity over the other two methods. SVM is a supervised learning model which requires training on the reference embeddings. Assuming there is a good SNN, it would seem redundant to train a model to separate the embeddings, since this is exactly what a SNN should already have done. Next, KNN has lower accuracy than cosine similarity. Therefor, we propose using the cosine similarity for classifying the embeddings during the inference stage.

5.4 Cow Identificaton

In its current state the SNN is not yet ready to be used in monitoring the drinking behaviour of cows, but it could be turned into a valuable model with some adjustments. As can be seen in figure 9, given only 26 images of the training data, the model can still predict the correct class with an accuracy of around 40%. With the backgrounds removed, the best accuracy from the best epoch reaches over 15%, which is twice as good as a randomly guessing model (with an expected accuracy of $\frac{1}{classes} = \frac{1}{13} = 0.077$). This can still be considered impressive since the model was trained on IC-64 without background removal.

5.5 FaceNet-like Application

It has already been shown that it is possible to identify cows using deep learning (Bello *et al.* 2020; Wang *et al.* 2014; Yao *et al.* 2019; Li *et al.* 2021). However, all previously developed models only perform on the herd they were trained on and thus require retraining when applying the model to a study on a different herd. Future research would benefit most from a model that can generalize well to all cows' spot patterns so expensive retraining can be eliminated. A SNN is a model that can potentially achieve such generalization. This has already been shown in the case of face-recognition by FaceNet (Schroff and Philbin 2015), although the size of their dataset was considerably larger. Consequently, training a SNN on the images from multiple herds combined could lead to a universally applicable model for cow identification. If our model can be implemented for individual FWI monitoring, multiple dairy farms could potentially be interested in using our software. At first, this will require training the SNN at each farm for their specific herd. Through federated learning, it could be possible to train one extra SNN by leveraging the data of all dairy farms that use our software combined. This can be a way of obtaining this universal FaceNet-like individual cow identifier.

5.6 FAIRness

In it's current state, the SNN cannot be used for identification in daily monitoring the FWI of individual cows. Changes have been suggested that can improve the model. The author will not be working on this project anymore as the time allocated has past. To ensure this work can be continued, several measures have been taken in accordance with the FAIR-principles (Wilkinson *et al.* 2016).

Code Reusability

All code was documented thoroughly and is made publicly available on GitHub. Additionally, Jupyter notebooks are provided in the same repository to demonstrate how the code can be used. Finally, all Python-packages that are required for running any of the code can be installed through the requirements-file, which is also available in the same repository.

Data Availability

The collected data has been stored on a external hard drive provided by the corresponding author and has been returned to the corresponding author upon expiration of the project. All data has been diligently structured and was provided with proper documentation and meta-data. Data can be made accessible upon request.

6 Conclusion

A thorough basis has been created for the development of a SNN that can be used to identify individual cows. The model needs to be improved much using instance segmentation to remove the image backgrounds. This can bring us a step closer towards a much needed individual FWI monitor for dairy farms to ensure that all cows' water needs are met, which is required for the animal health and welfare.

References

Ba, J. L., J. R. Kiros, and G. E. Hinton. 2016. “Layer Normalization” (July). https://doi.org/https://doi.org/10.48550/arXiv.1607.06450. http://arxiv.org/abs/1607.06450.

Barkema, H. W., M. A. von Keyserlingk, J. P. Kastelic, T. J. Lam, C. Luby, J. P. Roy, S. J. LeBlanc, G. P. Keefe, and D. F. Kelton. 2015. “Invited review: Changes in the dairy industry affecting dairy cattle health and welfare.” *Journal of dairy science* 98 (11): 7426–7445. ISSN: 1525-3198. https://doi.org/10.3168/JDS.2015-9377. https://pubmed.ncbi.nlm.nih.gov/26342982/.

Bello, R. W., A. Z. Talib, A. S. A. Mohamed, D. A. Olubummo, and F. N. Otobo. 2020. “Image-based individual cow recognition using body patterns.” *International Journal of Advanced Computer Science and Applications* 11 (3): 92–98. ISSN: 21565570. https://doi.org/10.14569/IJACSA.2020.0110311.

Cardot, V., Y. L. Roux, and S. Jurjanz. 2008. “Drinking behavior of lactating dairy cows and prediction of their water intake.” *Journal of Dairy Science* 91 (6): 2257–2264. ISSN: 15253198. https://doi.org/10.3168/jds.2007-0204.

Hafiz, A. M., and G. M. Bhat. 2020. “A survey on instance segmentation: state of the art.” *International Journal of Multimedia Information Retrieval* 9 (3): 171–189. ISSN: 2192662X. https://doi.org/10.1007/S13735-020-00195-X/TABLES/3.

Han, J., M. Kamber, and J. Pei. 2012. *Data Mining: Concepts and Techniques.* Elsevier Inc. ISBN: 9780123814791. https://doi.org/10.1016/C2009-0-61819-5.

Houpt, T. 1984. *Water balance and excretion.* 10th ed. Edited by S. M.J. 178–183. Cornstock Publishing Co.

Huang, T. S. 1996. “Computer Vision: Evolution and Promise,” https://doi.org/10.5170/CERN-1996-008.21.

Huuskonen, A., L. Tuomisto, and R. Kauppinen. 2011. “Effect of drinking water temperature on water intake and performance of dairy calves,” https://doi.org/10.3168/jds.2010-3723.

Jensen, M. B., and M. Vestergaard. 2021. “Invited review: Freedom from thirst—Do dairy cows and calves have sufficient access to drinking water?” *Journal of Dairy Science* 104 (11): 11368–11385. ISSN: 0022-0302. https://doi.org/10.3168/JDS.2021-20487.

Jocher, G., A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, *et al.* 2020. “ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements” (October). https://doi.org/10.5281/ZENODO.4154370.

Kertz, A. F., L. F. Reutzel, and J. H. Mahoney. 1984. “Ad Libitum Water Intake by Neonatal Calves and Its Relationship to Calf Starter Intake, Weight Gain, Feces Score, and Season,” https://doi.org/10.3168/jds.S0022-0302(84)81660-4.

Koch, G., R. Zemel, and R. Salakhutdinov. 2015. “Siamese Neural Networks for One-shot Image Recognition.”

Li, S., L. Fu, Y. Sun, Y. Mu, L. Chen, J. Li, and H. Gong. 2021. “Individual dairy cow identification based on lightweight convolutional neural network.” *PLOS ONE* 16 (11): e0260510. ISSN: 1932-6203. https://doi.org/10.1371/JOURNAL.PONE.0260510.

Liang, Y., R. Hudson, and M. Ballou. 2020. “Supplementing neonatal Jersey calves with a blend of probiotic bacteria improves the pathophysiological response to an oral Salmonella enterica serotype Typhimurium challenge,” https://doi.org/10.3168/jds.2019-17480. https://doi.org/10.3168/jds.2019-17480.

Lin, T. Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. 2014. “Microsoft COCO: Common objects in context.” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8693 LNCS (PART 5): 740–755. ISSN: 16113349. https://doi.org/10.1007/978-3-319-10602-1_48/COVER.

Little, W., K. A. Collis, P. T. Gleed, B. F. Sansom, W. M. Allen, and A. J. Quick. 1980. “Effect of reduced water intake by lactating dairy cows on behaviour, milk yield and blood composition.” *The Veterinary record* 106 (26): 547–551. ISSN: 0042-4900. https://doi.org/10.1136/VR.106.26.547.

Mathis, A., P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge. 2018. "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning." *Nature Neuroscience 2018 21:9* 21 (9): 1281–1289. ISSN: 1546-1726. https://doi.org/10.1038/s41593-018-0209-y.

Matthews, S. G., A. L. Miller, J. Clapp, T. Plötz, and I. Kyriazakis. 2016. "Early detection of health and welfare compromises through automated detection of behavioural changes in pigs." *Veterinary journal (London, England : 1997)* 217 (November): 43–51. ISSN: 1532-2971. https://doi.org/10.1016/J.TVJL.2016.09.005.

Melin, M., H. Wiktorsson, and L. Norell. 2005. "Analysis of Feeding and Drinking Patterns of Dairy Cows in Two Cow Traffic Situations in Automatic Milking Systems." *J. Dairy Sci* 88:71–85. https://doi.org/10.3168/jds.S0022-0302(05)72664-3.

Mendes, E. D., G. E. Carstens, L. O. Tedeschi, W. E. Pinchak, and T. H. Friend. 2011. "Validation of a system for monitoring feeding behavior in beef cattle." *Journal of animal science* 89 (9): 2904–2910. ISSN: 1525-3163. https://doi.org/10.2527/JAS.2010-3489.

Meyer, U., M. Everinghoff, D. Gädeken, and G. Flachowsky. 2004. "Investigations on the water intake of lactating dairy cows." *Livestock Production Science* 90 (2-3): 117–121. ISSN: 0301-6226. https://doi.org/10.1016/J.LIVPRODSCI.2004.03.005.

Murphy, M. R. 1992. "Water Metabolism of Dairy Cattle." *Journal of Dairy Science* 75 (1): 326–333. ISSN: 00220302. https://doi.org/10.3168/jds.S0022-0302(92)77768-6.

Nath, T., A. Mathis, A. C. Chen, A. Patel, M. Bethge, and M. W. Mathis. 2019. "Using DeepLabCut for 3D markerless pose estimation across species and behaviors." *Nature Protocols 2019 14:7* 14 (7): 2152–2176. ISSN: 1750-2799. https://doi.org/10.1038/s41596-019-0176-0.

Ridoutt, B. G., S. R. O. Williams, S. Baud, S. Fraval, and N. Marks. 2010. "Short communication: The water footprint of dairy products: Case study involving skim milk powder," https://doi.org/10.3168/jds.2010-3546.

Rousseeuw, P. J. 1987. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis." *Journal of Computational and Applied Mathematics* 20:53–65. ISSN: 0377-0427. https://doi.org/https://doi.org/10.1016/0377-0427(87)90125-7.

Schroff, F., and J. Philbin. 2015. "FaceNet: A Unified Embedding for Face Recognition and Clustering," https://doi.org/10.1109/CVPR.2015.7298682.

Steinfeld, H., P. J. Gerber, T. Wassenaar, V. Castel, M. Rosales, and C. D. haan. 2006. *Livestock's Long Shadow: Environmental Issues and Options.* Vol. 24. September. ISBN: 978-92-5-105571-7.

Szegedy, C., S. Ioffe, V. Vanhoucke, and A. Alemi. 2016. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," https://doi.org/https://doi.org/10.48550/arXiv.1602.07261.

Wang, J., Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. 2014. "Learning Fine-grained Image Similarity with Deep Ranking," https://doi.org/https://doi.org/10.48550/arXiv.1404.4661.

Wilkinson, M. D., M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, *et al.* 2016. "The FAIR Guiding Principles for scientific data management and stewardship." *Scientific Data 2016 3:1* 3 (1): 1–9. ISSN: 2052-4463. https://doi.org/10.1038/sdata.2016.18.

Yao, L., H. Liu, Z. Hu, Y. Kuang, C. Liu, and Y. Gao. 2019. "Cow face detection and recognition based on automatic feature extraction algorithm." *ACM International Conference Proceeding Series* (May). https://doi.org/10.1145/3321408.3322628. https://doi.org/10.1145/3321408.3322628.

Zehetmeier, M., J. Baudracco, H. Hoffmann, and A. Heißenhuber. 2012. "Does increasing milk yield per cow reduce greenhouse gas emissions? A system approach." *Animal* 6 (1): 154–166. ISSN: 1751-7311. https://doi.org/10.1017/S1751731111001467.