Computers and Electronics in Agriculture 155 (2018) 12–22

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Original papers

Body condition estimation on cows from depth images using Convolutional Neural Networks

Juan Rodríguez Alvarez[c,*], Mauricio Arroqui[a,c], Pablo Mangudo[a,c], Juan Toloza[a,c], Daniel Jatip[a,c], Juan M. Rodríguez[b], Alfredo Teyseyre[b], Carlos Sanz[d], Alejandro Zunino[b], Claudio Machado[a,c], Cristian Mateos[b]

[a] D-TEC – Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT), Argentina
[b] ISISTAN – UNICEN – CONICET, Tandil, Buenos Aires, Argentina
[c] CIVETAN – UNICEN – CONICET – CICPBA, Tandil, Buenos Aires, Argentina
[d] Faculty of Veterinary Sciences – UNICEN, Argentina

ARTICLE INFO

Keywords:
Precision livestock
Body condition score
Image analysis
Convolutional Neural Networks

ABSTRACT

BCS ("Body Condition Score") is a method used to estimate body fat reserves and accumulated energy balance of cows. BCS heavily influences milk production, reproduction, and health of cows. Therefore, it is important to monitor BCS to achieve better animal response, but this is a time-consuming and subjective task performed visually by expert scorers. Several studies have tried to automate BCS of dairy cows by applying image analysis and machine learning techniques. This work analyzes these studies and proposes a system based on Convolutional Neural Networks (CNNs) to improve overall automatic BCS estimation, whose use might be extended beyond dairy production.

The developed system has achieved good estimation results in comparison with other systems in the area. Overall accuracy of BCS estimations within 0.25 units of difference from true values was 78%, while overall accuracy within 0.50 units was 94%. Similarly, weighted precision and recall, which took into account imbalance BCS distribution in the built dataset, show similar values considering those error ranges.

1. Introduction

Body condition score (BCS) refers to the relative amount of subcutaneous body fat or energy reserve in cows, regardless of body weight and frame size (Wildman et al., 1982). BCS uses a 5-point scale system with 0.25-point increments, ranging from 1 representing emaciated cows, to 5 representing obese cows (Ferguson et al., 1994, 2006). A BCS is assigned to a cow based on the appearance of tissue cover over the bony prominences in the back and pelvic regions (Ferguson et al., 1994). BCS is an important management tool, which can improve herd nutrition, health, production, and pregnancy rate (Heinrichs et al., 2017; Kellogg, 2010; Markusfeld et al., 1997; Roche et al., 2009). However, BCS estimation is a time-consuming process measured manually by trained evaluators. The subjectivity in the judgment of evaluators can lead to different scores for the same cow under consideration, and could be influenced by previously observed cows (Bercovich et al., 2013). For instance, Ferguson et al. (1994) demonstrated that human observers agreed 58.1% of the time with a modal BCS of 4 observers, and varied by 0.25 and 0.50 units, 32.6% and 6.8% of the time respectively. Also, Ferguson et al. (1994) found that a less experienced observer (Ferguson et al., 1994) was within 0.25 units of the modal BCS 65% of the time and within 0.50 units 84.4% of the time. Thus, BCS changes of 0.25 cannot realistically be detected, even with trained observers (Bewley et al., 2008; Bewley and Schutz, 2008). Therefore, the increasing advances in technology availability at an accessible cost, automation, and digitalization of livestock farming tasks offer multiple opportunities. In this context, different studies have particularly focused on BCS automation (Shelley, 2016; Fischer et al., 2015; Hansen et al., 2015; Bercovich et al., 2013; Halachmi et al., 2013; Azzaro et al., 2011; Spoliansky et al., 2016).

However, according to the literature review detailed in the next Section, there is no unique system which achieves the following desirable features together:

1. uses images as the only information source (without external data such as weight, age, or lactation stage of the cow),
2. uses low-cost hardware resources,
3. gets real-time estimations,

* Corresponding author.
E-mail address: jrodriguezalvarez@alumnos.exa.unicen.edu.ar (J. Rodríguez Alvarez).

https://doi.org/10.1016/j.compag.2018.09.039
Received 18 December 2017; Received in revised form 10 September 2018; Accepted 25 September 2018

4. processes as accurately and automatically as possible.

Particularly, real-time evaluation does not represent a problem on dairy farm activities because farmers interact with the cows at least twice a day. However, it is very important in beef cow-calf operations, where interactions with cows have seasonal frequency, and contingency actions should be applied immediately (e.g. herd split) to avoid unnecessary herd movements. Additionally, it is important that the proposed system achieves accurate estimations by using a cheap camera, allowing its use to a wide variety of producers.

Therefore, the main objective of this study was to develop an *accurate*, *automatic*, *real-time,* and *low-cost* BCS estimation system, using the cow image as the only information source. A preliminary short version of this work was published in Rodríguez Alvarez et al. (2017). In accordance with the evolution of the research, a system based on state-of-the-art artificial intelligence techniques and the knowledge necessary to apply it in practice are described in detail in the present work. Additionally, a detailed experimental analysis is carried out to validate the proposed system.

2. Related works

Several attempts to automate the determination of dairy cows' BCS using digital images are reported in the literature. Table 1 shows a comparison among different works, according to the following aspects:

- Camera: type of camera used to capture cow images.
- Breed: cows breed used in the experiments, which include Holstein, SRB (Swedish Red Breed), and Fleckvieh.
- Dataset (images): amount of images used during the analysis.
- Automatization: system automation level. Considered values are low, medium, and high.
- Real-time: indicates if BCS estimations are available immediately after images are captured, or if time required by preprocessing tasks delays BCS estimations.
- Segmentation: indicates if image is segmented, i.e. if each cow is separated from the background image.
- Normalization: denotes if cow representation is normalized to make it independent of cow dimensions.
- Features: enumerates main cow features, derived from image analysis or obtained from external sources, which are used to feed the BCS estimation model.
- Model: type of model used to estimate BCS from identified features.
- Results: shows main obtained results. Among used metrics it is possible to identify percentage of error, correlation between true and estimated BCS, and model accuracy within different human error ranges (0.25, 0.50 and/or 0.75 units of difference between true and estimated BCS).

From the literature review, it follows that developed systems have broadly comprised two stages: (i) image analysis techniques to extract relevant characteristics (such as angles, distances, and areas between anatomical points; intensity/depth pixels values; cow contour or a representation of it) to differentiate fat reserve levels of cows; (ii) usage of collected characteristics to implement a BCS estimation model. Mostly, there are two types of models used: regression analysis models (as in Azzaro et al., 2011; Bercovich et al., 2013; Bewley et al., 2008; Fischer et al., 2015; Krukowski, 2009; Salau et al., 2014; Spoliansky et al., 2016) and algorithms that measure cow's body angularity (as in Halachmi et al., 2013; Hansen et al., 2015; Shelley, 2016) according to the hypothesis that the body shape of a fatter cow is rounder than that of a thin cow.

On the other hand, three automation levels exist. In the lowest level are (Azzaro et al., 2011; Bewley et al., 2008; Fischer et al., 2015), which require to manually identify anatomical points in the images to extract characteristics to develop the estimation models. In the medium level are (Anglart, 2010; Bercovich et al., 2013; Krukowski, 2009), where the input images used are manually selected, but the rest of the process is automatic. Finally, in the highest level are (Halachmi et al., 2013; Hansen et al., 2015; Salau et al., 2014; Shelley, 2016; Spoliansky et al., 2016), which achieve a completely automated process. Among the latter studies, only Halachmi et al. (2013) and Hansen et al. (2015) carry out real-time estimations because image preprocessing techniques (segmentation, normalization, features extraction) used in the other studies are time-consuming and therefore are performed under a batch scheme. However, Halachmi et al. (2013) use a very expensive thermal camera (in comparison with the other studies) and Hansen et al. (2015) do not perform a detailed analysis of results and only corroborate the inversely proportional relationship between BCS and angularity of the cow's body.

Summarizing, each one of the related works mentioned lacks any of the four desirable features for BCS systems pointed out in Section 1. In this sense, the application of a novel machine-supported learning model in the area helps us to improve estimations performance, with neither sophisticated preprocessing steps that negatively impact in real-time evaluations nor additional information of cows, such as age and weight, to feed the BCS estimation model. Additionally, the use of a inexpensive camera to acquire cow images reduces the cost of the system.

3. Material and methods

Motivated by the problems mentioned above, a new system to estimate BCS in real-time is proposed. The developed system relies on input images, which are collected to build a dataset of cows with their associated BCS values. Then, those images are used to train and validate a CNN model, a type of classifier which uses a special approach and a particular architecture to deal with pattern recognition and image classification problems. Fig. 1 overviews the developed system oriented towards estimating cow BCS values from captured images. The stages comprising this system will be described in the following subsections.

3.1. Data collection

To support both the development of the system and its validation, three dairy farms were visited to acquire the images. One of them (dairy farm number 1) is located in Carlos Pellegrini, Santa Fe (Argentina),[1] and has about 1000 Holstein cows. Another (dairy farm number 2) is located in Gardey, Buenos Aires (Argentina),[2] and has about 450 Holstein cows. The last one (dairy farm number 3) is located in Vela, Buenos Aires (Argentina),[3] and has about 250 Holstein cows. Fig. 1 shows the device used to capture the images as the cows walked by voluntarily below the camera (Microsoft Kinect v2 ToF). Interest in this type of camera for livestock application is increasing in term of its high quality and low cost (Marinello et al., 2015) (around U$S100). The Kinect v2 provides a big improvement over the original Kinect for outdoor/sunlight situations. While the original version is not suited to outdoor usage, the Kinect v2 device is less sensitive to daylight (Lachat et al., 2015). The Kinect v2 can be used outdoors in overcast situations as valid measurements can be acquired for ranges up to ~2.8 m. In direct sunlight, the data quality of the Kinect v2 strongly depends on the distance to the target and the incidence angle of the sunlight. Although valid measurements may be obtained with the Kinect v2 in direct sunlight for distances up to ~1.9 m, an increase in noise by approximately an order of magnitude should be expected (Fankhauser et al., 2015).

The device was used in the same way and under the same environment conditions across the three dairy farms, so as to delete or reduce external factor influence, and to build a unique whole dataset of

[1] https://goo.gl/maps/MASS59JYf6U2.
[2] https://goo.gl/maps/wfmCvKExpJD2.
[3] https://goo.gl/maps/3g1oW42vmvL2.

Table 1
Related works comparison.

Work	Camera	Breed	Dataset (images)	Automatization	Real- time	Segmentation	Normalization	Features	Model	Results
Bewley et al. (2008))	Digital, 2D	Holstein-Fresian	834 (US-BCS), 767 (UK-BCS)	Low	No	No	No	Angles between anatomical points	2 mixed models, one of them using additional angles	Acc: 92.79% within 0.25, Acc: 100% within 0.5
Krukowski (2009)	ToF, 3D	SRB	351 training, 120 test	Medium	No	Yes	Yes	Identification of anatomical points and calculation of derived values	LR: Linear Regression. PR: Polynomial Regression (quadratic terms)	Test Set (PR): Acc: 20% within 0.25, Acc: 46% within 0.5
Anglart (2010)	ToF, 3D	SRB	1329 (10% training, 90% test)	Medium	N/A	N/A	N/A	N/A: 3D information of part of the spinal transverse process	N/A: Neural Network	$R = 0.84$. Acc: 69% within 0.25, Acc: 95% within 0.50
Azzaro et al. (2011)	Digital, 2D	Holstein-Fresian	286	Low	No	No	Yes	Shape definition, and KPCA to get shape descriptors from an average one.	Principal Components Regression	$Error_{LOOCV} = 0.31$
Halachmi et al. (2013))	Termal	Holstein	172	High	Yes	Yes	No	Cow contour, fitted parabola	Function of deviation (MAE) of fitted parabola from cow contour (angularity)	$R = 0.94$
Bercovich et al. (2013)	Digital 2D	Holstein	87 training, 64 test	Medium	No	Yes	Yes	Distances and angles between anatomical points, and COW SIGNATURE.	Linear Regressions. Best model uses Fourier Descriptors of cow's signatures as prediction variables (FD-CW)	Test Set (FD-CW): $R^2 = 0.64$. Acc: 43% within 0.25, Acc: 72% within 0.50
Salau et al. (2014)	ToF 3D	Fleckvieh	540 (for GLM with all features), 514 (for correlation analysis on individual features)	High	No	Yes	No	Area and distances between body traits extracted from COW SIGNATURE, derived polynomial functions	FC: individual features correlation against manual BCS. GLM: Generalized Linear Model, using all features and lactation week	$R^2_{GLM} = 0.7$
Hansen et al. (2015)	*Light Coding*, 3D	Holstein-Fresian	95	High	Yes	Yes	Yes	Three-dimensional surface of body cow and fitted sphere	*Rolling Ball* (3D angularity)	N/A. High repeatability scoring an individual cow (14/15). Inverse relationship between angularity and BCS
Fischer et al. (2015)	*Light Coding*, 3D	Holstein	57 training, 25 testing set *cows*, 25 testing set *stage*	Low	No	No	Yes	PCA using x,y,z normalized pixels values	Principal Components Regression	Test dataset *stage* (cows of training set, but in different lactation stage): $R = 0.89$ and $RMSE = 0.31$. Test dataset *cows*(non seen cows): $R = 0.96$ and $RMSE = 0.32$
Shelley (2016)	*Light Coding*, 3D	Holstein	18517	High	No	Yes	No	Transverse cross-sections of cow body and fitted parabolas	Function of deviation (MAE) of fitted parabolas from cow contours (angularity)	Acc: 71.35% within 0.25, Acc: 93.91% within 0.5
Spoliansky et al. (2016)	*Light Coding*, 3D	Holstein	11824 training, 2650 test	High	No	Yes	Yes	Derived calculations over depth values (general and by zones to the image), distance from center to cow contour, weight, age	Polynomial Regression using quadratic terms	$R^2 = 0.75$. $MAE = 0.26$. Acc: 74% within 0.25, Acc: 91% within 0.5

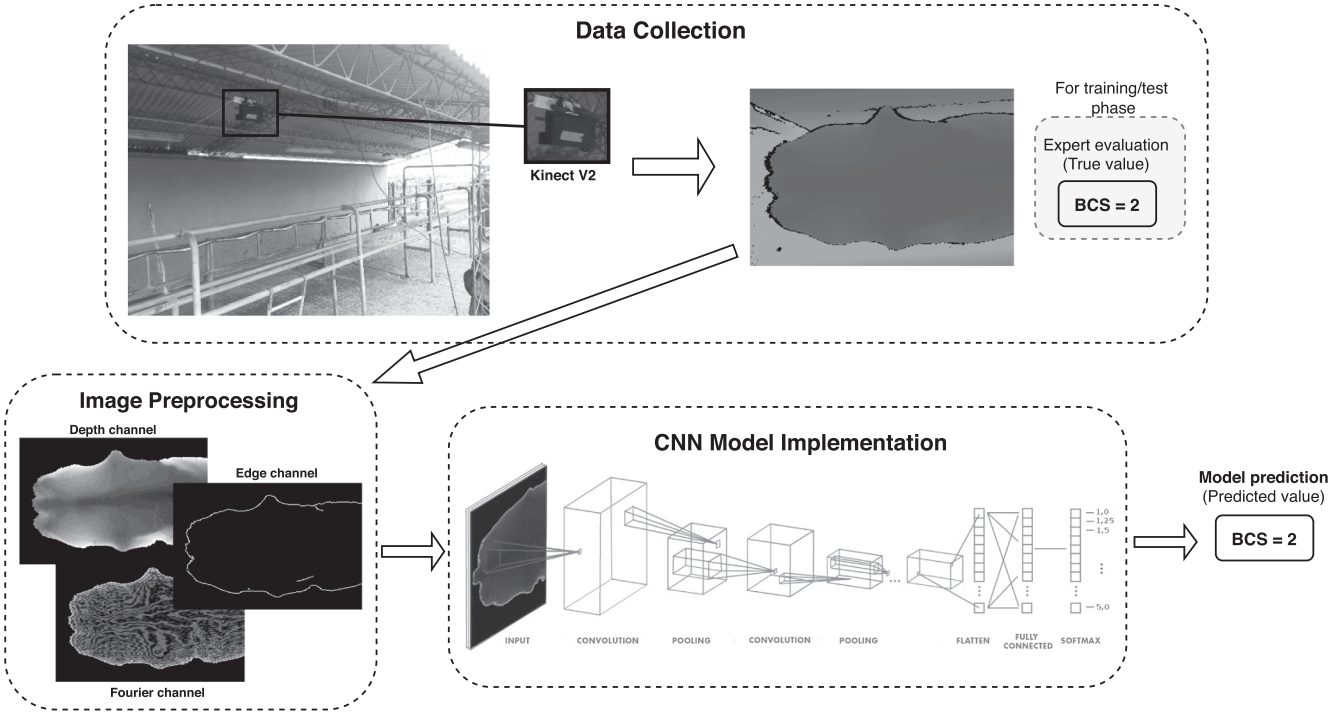


Fig. 1. Overview of developed BCS estimation system.

Table 2
BCS values distribution and their corresponding class weight.

BCS value	Distribution of training images	Distribution of test images	Classweight
1.75	4	2	24.12
2.00	35	15	2.75
2.25	86	37	1.12
2.50	270	117	0.36
2.75	205	89	0.47
3.00	207	89	0.47
3.25	148	64	0.65
3.50	127	55	0.76
3.75	49	22	1.97
4.00	20	9	4.82
4.25	5	3	19.30
4.50	2	1	48.25

images, regardless of capture locations. The device was located at the exit of the milk parlor, 2.8 m above ground, and aimed downward to an area that was not exposed to direct sunlight, in order to avoid the aforementioned Kinect problems and get valid pixel values. Depth 512 × 424 images were used to train/test the model because they have proven to be more suitable than RGB images to depict cow body variability associated with changes in BCS (Fischer et al., 2015).

During the acquisition of the cow images, an expert scorer evaluated the BCS of cows in situ so as to build a consistent labeled dataset. Cows were scored by the same expert in the three dairy farms in order to reduce subjectivity inconsistencies. An evaluator becomes an expert when he/she has learned how adipose tissues tend to accumulate on specific areas on the rump, loin, and ribs following well known guidelines, and has acquired enough practice to correctly assess and distinguish different BCS values.

The built dataset has 1661 cow depth images, all of them labeled by the expert scorer, which were split into training and test sets. In this sense, 70% of the images (1158) were used for model development (training) and the remaining images (503) were used for model validation. Both datasets were composed of BCS values ranging from 1.75 to 4.5 preserving samples distribution of the whole dataset, i.e. images were distributed proportionally into both datasets, as the first 3 columns of Table 2 show.

3.2. System development

This section describes the steps applied to process captured images (which involve segmentation, transformation, and normalization of depth images), the implementation of the learning model and the metrics selected to analyze its quality.

3.2.1. Image preprocessing

A Python script was implemented to automatically preprocess captured images. Well-known Python modules such as OpenCV,[4] Numpy,[5] and Scikit-Image[6] were used to process and manipulate those images, according to the algorithmic steps described next. First of all, a segmentation between background objects and the cow in the image was applied to filter pixels that do not belong to a cow's body. To do this, a capture of the empty scene was set as background image and then subtracted to cow images. Thus, only a cow's back end that was not present in the background image was conserved. A threshold was used on the background removal process to filter pixels with very similar depth values. That is, distances of 2 cm. or less between pixel values of cow and background image were considered virtually as the same value. Those pixels were removed from the cow image, because this difference could be produced by minor changes on camera measures. Additionally, pixels located above 1.8 m from the floor were filtered out, assuming that there are no cows taller than this value and therefore those pixels were irrelevant. Depth values were rescaled from 0 to 255 (8 bits representation) highlighting cow body variability, and making them independent of animal size. The next source code outlines the steps described above:

[4] https://docs.opencv.org.
[5] www.numpy.org.
[6] http://scikit-image.org.

```
def substract_empty_image(image_cow, empty_image, threshold
    ):
    image_dif = image_cow - empty_image
    image_dif = numpy.absolute(image_dif)
    img = numpy.where(image_dif<=threshold, 0, image_cow)
    return img.astype(dtype=numpy.uint8)

def post_delete_pixels_out_regular_distances(image,
    max_height):
    min_pixel_value = (((2.8-max_height)*1000.0)/4500.0)
        *255
    img = numpy.where(image<min_pixel_value, 0, image)
    return img.astype(dtype=numpy.uint8)

def scaling_depth(image):
    min = 0
    if image[image>0].size:
        min = numpy.min(image[image>0])
    max = numpy.max(image)
    img = image.astype(dtype=numpy.float64)
    img = numpy.where(img==0, 0, (img - min) / (max - min)
        * 255)
    return img.astype(dtype=numpy.uint8)

def delete_background(image_cow, empty_image):
    img = substract_empty_image(image_cow, empty_image,
        threshold = 2)
    img = post_delete_pixels_out_regular_distances(img,
        max_height=1.8)
    img = scaling_depth(img)
    return img.astype(dtype=numpy.uint8)
```

The resulted depth image was transformed generating 2 additional channels. One of them used Fourier transform to perform filtering operations to adjust the spatial frequency content of the depth image. To do this, firstly a Fourier transform was used to find the frequency domain of the depth image. Secondly, the transformed image was manipulated by applying a high-pass filtering, preserving only the higher spatial frequency components. Lastly, an inverse Fourier transform was performed to produce the final filtered image for the new channel, which preserves all of the sharp crisp edges from the original depth image. The other channel was generated using the Canny algorithm (Canny, 1986), an edge detector method used to locate sharp intensity changes and to find object boundaries (Ding and Goshtasby, 2001), which allowed for the cow body contour to be highlighted.

Summarizing, in the image preprocessing phase, cow images were segmented from the background and they were converted to a new one composed of 3 channels: depth, fourier transformation, and edge respectively. The following pseudocode summarizes all the steps described in this subsection:

```
def img_preprocessing(image_cow, empty_image):
    d = delete_background(image_cow, empty_image)
    f = fourier_transf_without_background(d)
    e = edges(d)
    shape = (d.shape[0], d.shape[1], 1)
    return np.concatenate((d.reshape(shape), f.reshape(
        shape), e.reshape(shape)), axis=-1)
```

3.2.2. Model implementation

A Convolutional Neural Network (CNN) was used to develop the model of this work. CNN is a powerful machine learning technique from the field of deep learning, which has not been exploited for BCS estimations. CNNs (Bengio et al., 2015) have been found highly effective and been commonly used in computer vision and image classification (Deng and Yu, 2014; Hijazi et al., 2015; Krizhevsky et al., 2012; Szegedy et al., 2015). A CNN is a specialized kind of neural network with a special architecture composed of a sequence of layers. Three main types of layers are used to build a CNN:

- Convolutional: captures visual patterns within an image by applying a specified number of filters to the image. Filters scan the input by subregions and perform a set of mathematical operations (convolutions) to generate convolved outputs (or feature maps) that respond to a visual element existing in the input. Each filter determines which feature the convolution is looking for. The idea is to generate different feature maps, each locating a specific simple characteristic in the image.
- Pooling (or subsampling): performs a downsampling operation along the spatial dimensions (width, height) of convolutional layers, to reduce the dimensionality of feature maps. A commonly used pooling algorithm is max pooling, which considers patch of the feature map (e.g., 2 × 2-pixels), keeps their maximum value, and discards the remaining values.
- Fully-connected: performs classification on the features extracted by the convolutional and pooling layers, connecting every node in this layer to every node in the preceding layer.

Additionally, a final layer with a softmax function is frequently used as the output of a classifier in order to represent a probability distribution over a discrete variable with “n” possible values (classes). In other words, it returns the estimated probability of each class, given a concrete sample.

In the traditional model of pattern/image recognition (studies of Table 1) a hand-designed feature extractor gathers relevant information from the input image. Then, features are used to train a classifier (or a regression model), which outputs the class (or value) corresponding to an input image. In a CNN, convolution and pooling layers play the role of feature extractor, where the weights (model coefficients or parameters) of the convolutional layer being used for feature extraction as well as the fully connected layer being used for classification are automatically determined during the training process (Hijazi et al., 2015). In this way, a CNN transforms the original image layer by layer from the original pixel values to the final class scores (the discrete BCS values within the 5-point scale).

In particular, SqueezeNet (Iandola et al., 2016) was used to implement the CNN model of this work. SqueezeNet is a small CNN architecture that achieves AlexNet (Krizhevsky et al., 2012) level accuracy on ImageNet with 50 times less parameters. Given equivalent accuracy, a CNN architecture with fewer parameters has several advantages. One of them is related to the shortest time required to train and evaluate a new model, taking advantage of the limited underlying computational resources available. For example, the developed model comprises 741580 parameters and uses around 2 GB of memory to train a batch of 16 images. This reduced memory consumption makes it possible to train a whole batch of images in a GPU with a reduced memory capacity, thus speeding up training by a considerable factor (often 5x to 10x, when moving from executing on a modern CPU to a single modern GPU). A GPU can perform lots of simple numerical processing tasks at the same time (massive parallelization), such as the huge amount of matrix multiplications and other relevant operations associated to a CNN training.

SqueezeNet is comprised mainly of “Fire” modules and uses 3 main strategies for reducing parameter size while maximizing accuracy: replaces 3 × 3 convolutional filters with 1 × 1, decreases the number of input channels to 3 × 3 filters, and delays downsampling closer to the output in the network so that convolution layers have large activation maps. This latter strategy is based on the intuition that large activation maps can lead to higher classification accuracy. Thus, strategies 1 and 2 aim to decrease the number of parameters in a CNN while attempting to preserve accuracy, and strategy 3 aims to maximize accuracy on a limited budget of parameters. “Fire” modules are composed of a convolution layer of 1 × 1 filters (called squeeze layer) followed by another convolution layer that has a concatenation of 1 × 1 and 3 × 3 filters (expand layer). Fig. 2 shows the CNN architecture used in this work.

Preprocessed images were used as CNN input values. Before feeding the network with an image, each image channel (Depth, Fourier Transformation, Edges) were zero-centered and normalized, scaling all pixel values between −1 and 1. Since CNNs work better with a huge number of training images, data augmentation techniques were applied in order to increase the number of image samples during the training phase. Data augmentation is a simple, commonly used technique to reduce overfitting on image data (Krizhevsky et al., 2012) and make a model generalize better. It works by creating fake data and adding it to the training set. This technique has been particularly effective for image classification problems, in which operations like translating the training

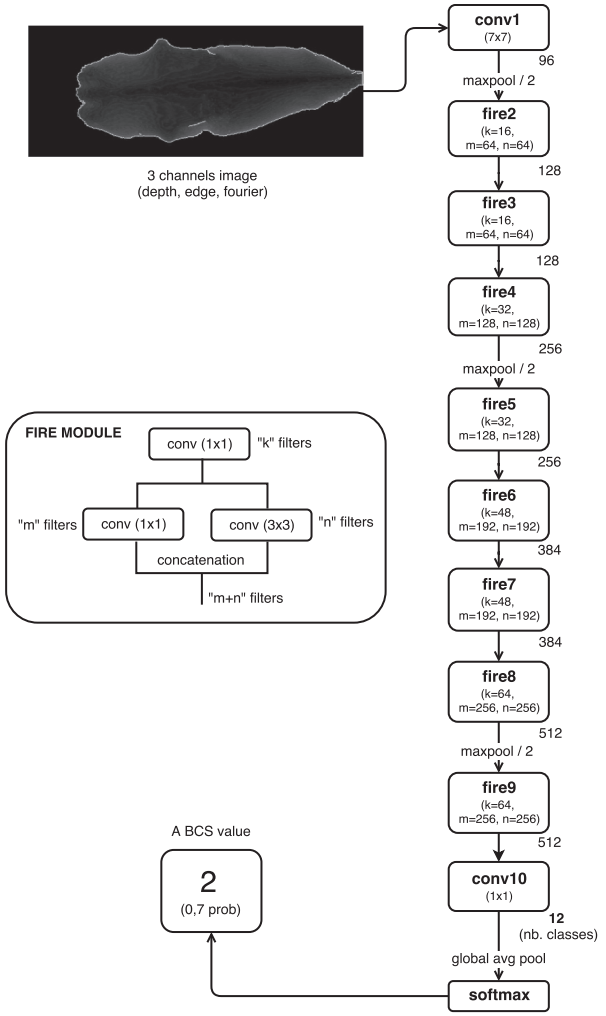


Fig. 2. CNN model architecture (SqueezeNet (Iandola et al., 2016)). Description of SqueezeNet "Fire" module, and structure of the CNN model from its input (preprocessed image) to the final predicted class or BCS value (class with highest probability according to the input image).

images a few pixels in each direction can improve generalization, even if the model has already been designed to be partially translation invariant. Many other operations such as rotating images or scaling images have also proven quite effective (Bengio et al., 2015). Particularly in this approach, each training image was modified by randomly rotating and flipping it, before presenting the image to the network. These affine transformations were employed because it is important not to modify image semantics during data augmentation, which could hinder learning. For example, distortions or isolated intensity pixel changes cannot been applied because they would modify body proportions associated with BCS changes.

In practice, building a dataset of cow images with an equitable distribution of BCS values is very difficult. A properly managed and feeding herd should not have extreme BCS values (Kellogg, 2010). That is why these values are less frequent and most cows on farms have a BCS value between 2 and 4 (Ferguson et al., 1994). For that reason, a strategy to deal with an imbalanced dataset had to be considered. Particularly, a technique which adjusts the importance of BCS classes was used, increasing the minority class weights during the training phase. Thus, the classifier will try to minimize the error for classes with higher "class weight", since errors of minority classes are considered more costly than those of the other classes. Table 2 shows sample distribution for each BCS value and their corresponding class weight, showing this imbalance and how specialized class weights were computed.

3.2.3. Development tools

The BCS estimation software was entirely written in Python, including the image preprocessing module and the implementation of the BCS estimation model. OpenCV, Numpy and Scikit-Image were used to develop the image preprocessing task, since these libraries provide well-known, highly-proven image processing algorithm implementations. Keras (Chollet et al., 2015) was used to develop the image classifier model using CNN. Keras is a model-level library that provides high-level building blocks for developing deep learning models, and works on top of Theano (Theano Development Team, 2016) or Tensorflow (Abadi et al., 2015). These frameworks serve as "backend engines" of Keras.

Keras models can run on GPUs using cuDNN (Chetlur et al., 2014) for high-performance GPU acceleration. cuDNN (part of the NVIDIA Deep Learning SDK) is a GPU-accelerated library that provides highly tuned implementations for standard CNN routines such as forward and backward convolution, pooling, normalization, and layer activation. The possibility of running models on GPU is particularly important and profitable during the training phase, when a huge number of images have to be processed and model weights have to be learned, involving compute-intensive tasks costly in time. Once the model has been developed, GPU acceleration is not so necessary, since only a set of known mathematical operations has to be applied over new input images, one by one.

3.3. Statistical analysis

A set of metrics was used to evaluate the CNN model, measuring the classification performance.

First of all, a confusion matrix was built because it is commonly used in classification problems (Bercovich et al., 2013; Chawla et al., 2002; Maimon and Rokach, 2010). The confusion matrix is a useful tool for analyzing how well a given classifier can recognize tuples of different classes showing a detailed breakdown of correct and incorrect classifications for each class. In this sense, it provides more information than a single value score, such as accuracy or precision, helping the assessment of the classification method (Stehman, 1997). In a confusion matrix, columns represent predicted classes and rows represent true classes, i.e. an entry *"row,column"* in a confusion matrix indicates the number of tuples of class *"row"* that were labeled by the classifier as class *"column"* (Han et al., 2011). Thus, the main diagonal of a confusion matrix shows the number of observations that have been correctly classified, while the off diagonal elements indicate the number of observations that have been incorrectly classified (Maimon and Rokach, 2010). In fact, for an individual class it is possible to identify four possible values: the number of correctly recognized class examples (tp = true positives), the number of correctly recognized examples that do not belong to the class (tn = true negatives), and examples that were either incorrectly assigned to the class (fp = false positives) or not recognized as class examples (fn = false negatives) (Sokolova and Lapalme, 2009).

Then, using the information of confusion matrix, the following measures were calculated (Sokolova and Lapalme, 2009):

- Accuracy: effectiveness of a classifier, that is the percentage of samples correctly classified.

$$Accuracy = \frac{\#correctPredictions}{\#predictions}$$

- Precision: ability of the classifier not to label a negative example as positive.

$$Precision = \frac{\sum_{i}^{\#classes}\left(\frac{tp_i}{tp_i + fp_i}\right)}{\#classes}$$

- Recall (a.k.a. sensitivity): ability of the classifier to find all the positive samples.

$$Recall = \frac{\sum_{i}^{\#classes}\left(\frac{tp_i}{tp_i + fn_i}\right)}{\#classes}$$

- F1-score: one measure that combines the trade-offs of precision and recall, and outputs a single number reflecting the "goodness" of a classifier in the presence of rare classes (Maimon and Rokach, 2010). It is the harmonic mean of precision and recall.

$$F1score = \frac{\sum_{i}^{\#classes}\left(2\frac{Precision_i * Recall_i}{Precision_i + Recall_i}\right)}{\#classes}$$

Accuracy was micro-averaged, i.e. it was overall assessed over the test data considering the number of correct predictions over the total number of test samples; whereas precision, recall, and F1-score were macro-averaged (average per-class measure), where metrics were calculated for each class and then values were weighted and unweighted average.

Additionally, for all calculated measures, classifications within human error ranges were taken into account, that is 0.25 and 0.50 units of differences between the true BCS value (ground truth) and predicted BCS value. Assessments within these ranges are frequently used in the literature to evaluate the accuracy of the models. Thus, the obtained results could be contrasted against other studies.

4. Results and discussion

The cares taken into account during data collection stage, that is, adopted restrictions related to device location in the field, have allowed to acquire a large number of valid images from different dairy farms. This enabled not only to easily scale the capture process, but also to contribute to model generalization (in combination with preprocessing tasks), making the learning process independent of a particular location. In addition, it is important to highlight the difficulties to obtain extreme BCS values, as the first 3 columns of Table 2 show. Therefore it was necessary to ensure that both training and test sets had a proportional distribution of samples to allow for a proper learning of the model and get suitable prediction values. In this sense, the technique used to adjust the importance of BCS classes was also very helpful.

Fig. 3 shows an example of images involved throughout the preprocessing steps applied to a single cow image. Fig. 3a and b show original depth images captured by the Kinect device. The first one corresponds to a capture of cow leaving the milk parlor, and the second corresponds to the empty scene used to remove background information. Fig. 3c, d, and e show the three channels generated by the preprocessing stage. All these figures show how the background was successfully removed using the operations described in Section 3.2.1 and how extra information was generated and combined to assist the learning process.

Fig. 4a and b show how data augmentation enables to train a suitable model by improving its generalization capabilities. These figures display the cross entropy loss (chosen error measure) as a function of training cycles number (epochs). Fig. 4a shows how increasing the number of images in the training set via data augmentation generates a better generalized model, i.e. a model which performs well on previously unobserved images and hence makes test error as low as possible. In this case, both training and test set error decrease as the number of epochs increases. On the other hand, Fig. 4b considers a training set without data augmentation. In this case, the test set error increases while the training error decreases, i.e. the model learns particularities of, or specializes, the training set, but fails to learn how to generalize predictions. This is known as overfitting. In brief, the predictions of the model trained without data augmentation are not useful for determining cows BCS.

The confusion matrix of test set samples classification is presented in Table 3. Ideally, most of the tuples would be represented along the diagonal of the confusion matrix with the rest of the entries being close to zero. Although this ideal state was not completely reached, it is possible to see that mostly samples per class are close to the diagonal, except for the highest BCS values, which shows a tendency in the identification of patterns to distinguish cow variability associated with changes in BCS. Problems in higher class classification were associated to low sample distribution of extreme BCS values, because it was difficult for the model to extract or learn patterns associated to these values considering only few training samples.

Using confusion matrix values, the global accuracy of the CNN model was calculated considering correct classifications within different human error ranges (ER). The following values were gotten:

- Accuracy (ER = 0, exact) = 40%
- Accuracy (ER = 0.25 BCS units) = 78%
- Accuracy (ER = 0.50 BCS units) = 94%

Table 4 shows precision, recall, and F1-score evaluations per class or BCS values in the test set. The two final table rows combine per class results to respectively calculate weighted and unweighted average metrics, i.e. these final rows present the macro-averaged classification measures of the model, considering (or not) the distribution of BCS values in the test set. Weighted metrics was added because, as it was shown before, dataset was imbalanced.

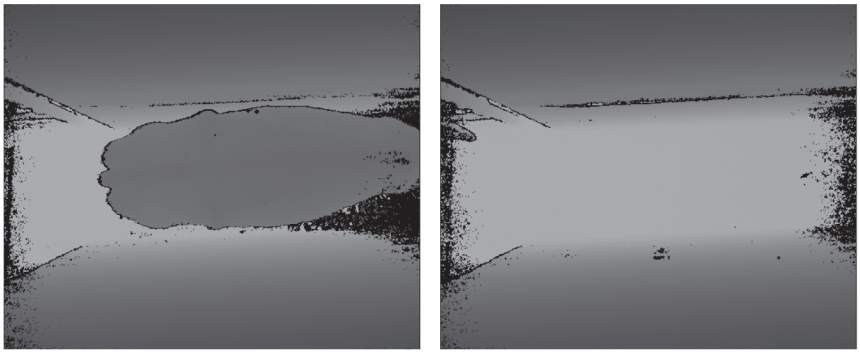
Tables 5,6 show the same measures as Table 4, but their values were calculated taking into account different human error ranges. Table 5 considers classifications within 0.25 units of error between true and predicted BCS value, while Table 6 considers classifications within 0.50 units of difference.

Tables show zero values for a metric when there are not true positive values for a class. Particularly, it is possible to see that BCS = 4.5 class could not be predicted by the model irrespectively of error range. This happened because the whole dataset of images had very few samples of this class (3 in all), because of which only two samples were used to train the model to identify particular patterns, and only one sample to test them. Table 4 also presents a not defined value (N/A) for precision of BCS = 4 class. This happens when the number of true positive plus false positive tuples for a class is zero. This is the case when the denominator of the precision equation is 0, i.e., the class was never predicted.

Overall accuracy and macro-averaging precision are some of the most frequently used metrics to evaluate classification models, but in this context, the recall metric is also very important to measure the quality of the model. For example, it is very important to achieve a good recall on extreme values, identifying all the occurrences of those values to prevent cow management or health problems, allowing for practitioners to take contingency actions.

Figs. 5–7 show classifier metrics (precision, recall, and F1-score respectively) in a bar chart representation, where each bar (x-axis) represents a BCS value or class, and the height of bar (y-axis) represents the metric results. Each bar is represented as a stack of 3 different gray scale colors (dark gray, gray, and light gray) representing metric values within different error ranges, as shown in legends.

Fig. 7 shows F1-score per class –which essentially combines precision and recall values into one single value– and highlights model difficulties to assess images on extreme BCS values. In this way, classes with more images to train, in the middle of the scale, present the best



(a) Original depth cow image (b) Original depth empty image

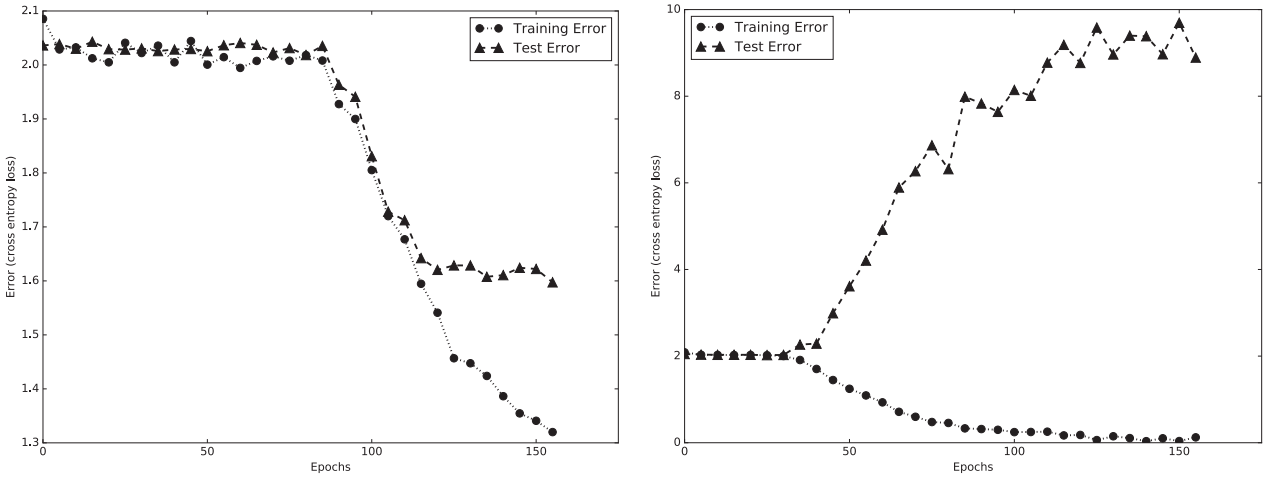
(c) Preprocessed image: depth channel (d) Preprocessed image: fourier channel (e) Preprocessed image: edge channel

Fig. 3. Example of preprocessed image: steps.

results. This could be improved by collecting not only more images in general, but also increasing the number of samples of extreme BCS values.

Also, as Ferguson et al. (1994, 2006) mentioned in their previous studies, it is difficult to distinguish BCS values less than 2 and greater than 4. In this sense, it may be advisable to modify the possible classes to recognize, keeping resolution degree (number of classes) in the middle of the scale ($2 \leqslant BCS \leqslant 4$), and grouping all BCS values less than 2 into one class (representing too thin cows), and all BCS values greater than 4 in another class (representing too fat cows).

Finally, overall model accuracies were contrasted against works presenting medium or high automatization level. Besides, as the BCS estimation problem was cast as a classification problem with discrete values, works which compute only metrics suitable for regression analysis and continuous values (such as R, R^2, $RMSE$) were not taken into account. Table 7 shows overall accuracy level achieved by related works and the developed system within different human error ranges, which is one of the most frequently used measure in the literature to evaluate the precision of models. However, it is important to note that it is not always possible to make a faithful comparison among models in the research area, since each study constructs its own dataset, usually private. In other words, there is no universal dataset of cow images that allows for a standardization of experimental factors, such as type of breed, scores distribution, and influence of experts' evaluations, which would enable to abstract model comparisons of variable factors. In this sense, only a high-level comparison could be made (similar to those



(a) Training set with Data Augmentation (b) Training set without Data Augmentation

Fig. 4. Training and test set errors as a function of the number of training cycles.

Table 3
Confusion matrix of test set samples classification. Dark gray cells represent exact predictions, gray cells represent predictions with 0.25 units of error, and light gray cells represent predictions with 0.50 units of error.

True Class \ Predicted Class	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	3.75	4.00	4.25	4.50
1.75	0	2	0	0	0	0	0	0	0	0	0	0
2.00	1	6	3	4	1	0	0	0	0	0	0	0
2.25	0	3	8	18	3	5	0	0	0	0	0	0
2.50	1	2	13	60	18	23	0	0	0	0	0	0
2.75	0	0	3	31	30	19	4	2	0	0	0	0
3.00	0	0	0	14	11	48	13	3	0	0	0	0
3.25	0	0	0	6	7	17	26	8	0	0	0	0
3.50	0	0	0	0	6	9	18	19	3	0	0	0
3.75	0	0	0	0	0	3	4	10	4	0	0	1
4.00	0	0	0	0	0	0	2	6	1	0	0	0
4.25	0	0	0	0	0	0	1	1	0	0	1	0
4.50	0	0	0	0	0	0	1	0	0	0	0	0

Table 4
Classification measures for exact predictions.

BCS value	Precision	Recall	F1-score
1.75	0.00	0.00	0.00
2.00	0.46	0.4	0.43
2.25	0.30	0.22	0.25
2.50	0.45	0.51	0.48
2.75	0.39	0.34	0.36
3.00	0.39	0.54	0.45
3.25	0.38	0.41	0.39
3.50	0.39	0.35	0.37
3.75	0.50	0.18	0.27
4.00	N/A	0.00	0.00
4.25	1.00	0.33	0.5
4.50	0.00	0.00	0.00
Weighted average per class	**0.40**	**0.40**	**0.39**
Unweighted average per class	**0.35**	**0.27**	**0.29**

Table 5
Classification measures within 0.25 range error.

BCS value	Precision	Recall	F1-score
1.75	0.67	1.00	0.80
2.00	0.83	0.67	0.74
2.25	0.91	0.78	0.84
2.50	0.79	0.78	0.78
2.75	0.82	0.90	0.86
3.00	0.64	0.81	0.72
3.25	0.81	0.80	0.80
3.50	0.77	0.73	0.75
3.75	1.00	0.64	0.78
4.00	1.00	0.11	0.20
4.25	1.00	0.33	0.50
4.50	0.00	0.00	0.00
Weighted average per class	**0.79**	**0.78**	**0.77**
Unweighted average per class	**0.77**	**0.63**	**0.65**

found in related works (Azzaro et al., 2011; Bercovich et al., 2013; Halachmi et al., 2013; Spoliansky et al., 2016)), which, as reported in Table 7, shows that the developed system in this work has achieved good results, thus overcoming in all cases accuracy estimations within 0.25 units of difference between true and predicted BCS value.

A last and additional analysis was made to demonstrate the system independence to particular dairy farm conditions. To do this, the general method developed in this study was used to train 3 different models. Each one of these models used two of the experimental sites to train (considering different combinations) and the other one as test set, in order to measure model predictions accuracy and generalization. Table 8 shows the results obtained considering the micro-average accuracy of the models within different human error ranges, where each model was named after the set of images used to test it as follow:

- General model, mixed test set (MT): the main model described and analyzed early, which combined images of the three sites in the training (70%) and test (30%) set.
- Test 1: model which used images from dairy farm number 1 as test set (1000 images), and the images of the other sites to train (661 images).

Table 6
Classification measures within 0.50 range error.

BCS value	Precision	Recall	F1-score
1.75	0.67	1.00	0.80
2.00	1.00	0.93	0.97
2.25	1.00	0.86	0.93
2.50	0.95	0.99	0.97
2.75	0.93	0.98	0.95
3.00	0.92	1.00	0.96
3.25	0.94	0.91	0.92
3.50	0.94	0.89	0.92
3.75	1.00	0.82	0.90
4.00	1.00	0.78	0.88
4.25	1.00	0.33	0.50
4.50	0.00	0.00	0.00
Weighted average per class	**0.94**	**0.94**	**0.94**
Unweightedaverage per class	**0.86**	**0.79**	**0.81**

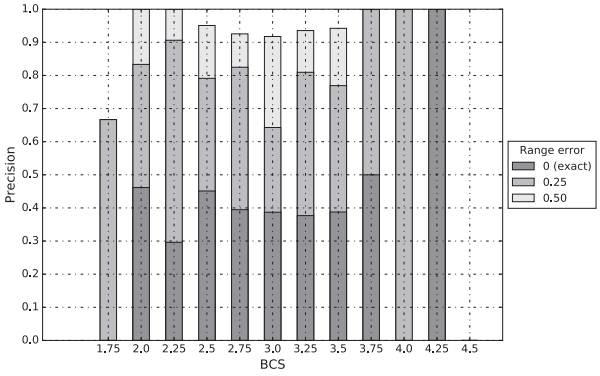


Fig. 5. Precisions per class according to different error ranges.

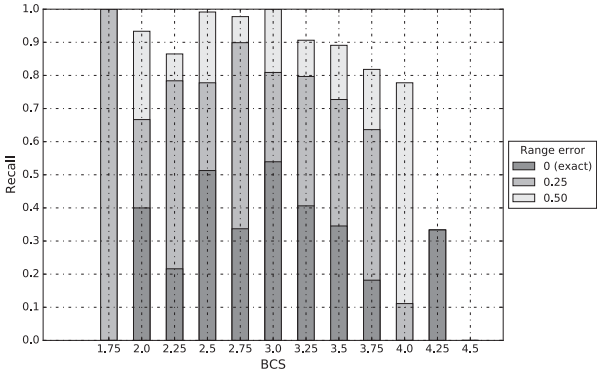


Fig. 6. Recalls per class according to different error ranges.

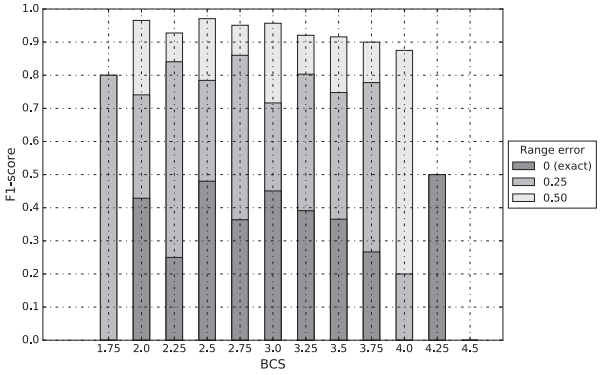


Fig. 7. F1-scores per class according to different error ranges.

Table 7
Overall accuracy level reported by related works and the developed system within different human error ranges.

Error range	Krukowski (2009)	Anglart (2010)	Bercovich et al. (2013)	Shelley (2016)	Spoliansky et al. (2016)	Developed CNN model
0.25	20%	69%	43%	71.35%	74%	**78%**
0.50	46%	95%	72%	93.91%	91%	**94%**

Table 8
Micro-average accuracy within different human error ranges, taking into account different test sets combinations.

	Models			
Error	Mixed Test	Test 1	Test 2	Test 3
0	0.40	0.31	0.35	0.36
0.25	0.78	0.73	0.77	0.77
0.50	0.94	0.92	0.94	0.95

- Test 2: model which used images from dairy farm number 2 (411 images) as test set, and the remaining images (1250) as training set.
- Test 3: model which used images from dairy farm number 3 (250 images) as test set, and the remaining images (1411) as training set.

Test 1 model presents the worst results because it used a small portion of data to train, which complicated the learning process in order to learn features or weights which would be used to recognize the BCS of unobserved images. However, the results obtained by Test 2 and Test 3 models have demonstrated that the considerations taken into to account to eliminate the influence of external factors related to a particular location were very useful to generalize model predictions.

5. Conclusions and future works

While some automatic methods to estimate BCS are available, new development opportunities have been identified to implement an accurate, automatic, real-time, and low-cost BCS estimation system, allowing for its application beyond dairy production. The cornerstone of this system is CNN, an effective technique to classify images, which has not been exploited for the classification problem at hand yet. Reported experiments confirm that using CNNs improves BCS estimations within 0.25 units of accuracy in comparison with previous works, and has achieved comparable results within 0.50 BCS units.

Further works should increase the dataset of images. To date, it is composed by 1661 depth images, with few samples on extreme BCS values. The number of images necessary to get good estimations results will depend on:

- the CNN design and configuration, or hyperparameters of the model, i.e. settings that can be used to control the behavior of the learning algorithm, which are not adapted by the learning algorithm itself (Bengio et al., 2015); and
- the difficulty of the learning problem, i.e. correctly distinguishing BCS values.

A rough rule of thumb is that a supervised deep learning algorithm will generally achieve good performance with around 5000 labeled examples per category (Bengio et al., 2015). However, according to partial obtained results, a dataset of around 1000–2000 images per class (each possible BCS value) in combination with data augmentation techniques (a way to artificially expand a dataset, by applying transformations to training data) should be suitable to achieve results close to the best possible.

The built dataset used to train and validate the system was composed of a mix of images of the three dairy farms. A set of considerations have been taken into to account to eliminate or reduce external factor influences related to a particular location. In the same way, cow segmentation from background objects contributes to the model independence of particular non-cow elements present in the images. Moreover, as the number of captures from different dairy farms increases, the generalization of the model improves completely ignoring non-cow factors (possible noise introduced by background objects, careless device installation, etc.) which negatively affect the learning process.

The system developed in this study, like the surveyed studies, could not been evaluated over different breeds (all experiments were made using Holstein cows only). In this sense, in future works it would be interesting to adapt the proposed system to learn different cow breeds particularities, in order to build a general method and deep learning architecture suitable for multi-breed evaluations. One of the hypothesis to validate is whether following the cares and steps previously described in Sections 3.1 and 3.2.1, and training the developed model over a new set of images to adapt its weights to the breed of interest, is enough to generate a fitted model to a particular breed. In other words, whether regardless of the cow breed considered in the experiments, the method and model architecture to be used would be the same, and only the weights of the model should be fitted by re-training it using new breed images.

Also, different CNN architectures and hyperparameters configurations should be considered, in search of the best performance of the model. For example, instead of thinking of this problem as a classification one, it could be treated as a regression problem by using continuous BCS values, as it was treated in some related works (Krukowski, 2009; Azzaro et al., 2011; Fischer et al., 2015). To do that, a proper cost function (e.g. Mean Square Error or Mean Absolute Error) should be used and the CNN architecture should be adapted to provide a continuous output, replacing the output layer by another of size 1 with an activation function suitable to a regression problem.

Additionally, it is important to consider that when the BCS system starts to work in a farm, a huge number of images and body condition values will be periodically generated. This amount of data could be analyzed individually or together with other information sources, such as sensors around the farm and on animals (such as an activity meter collar), local and external information systems, and custom digital registers. Thus, Big Data techniques (Bazán-Vera et al., 2017) could be applied to organize, analyze, process, and interpret such large volumes of diverse data, in an integrated system which could suggest contingency actions or automatically react to particular events.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org. <https://www.tensorflow.org/>.

Anglart, D., 2010. Automatic estimation of body weight and body condition score in dairy cows using 3d imaging technique (Master's thesis). Faculty of Veterinary Medicine and Animal Science, Swedish University of Agricultural Sciences.

Azzaro, G., Caccamo, M., Ferguson, J., Battiato, S., Farinella, G., Guarnera, G., Puglisi, G., Petriglieri, R., Licitra, G., 2011. Objective estimation of body condition score by modeling cow body shape from digital images. J. Dairy Sci. 94 (4), 2126–2137. https://doi.org/10.3168/jds.2010-3467.

Bazán-Vera, W., Bermeo-Almeida, O., Samaniego-Cobo, T., Alarcon-Salvatierra, A., Rodríguez-Méndez, A., Bazán-Vera, V., 2017. The Current State and Effects of Agromatic: A Systematic Literature Review. Springer International Publishing, Cham.

Bengio, Y., Goodfellow, I.J., Courville, A., 2015. Deep learning. Nature 521, 436–444.

Bercovich, A., Edan, Y., Alchanatis, V., Moallem, U., Parmet, Y., Honig, H., Maltz, E., Antler, A., Halachmi, I., 2013. Development of an automatic cow body condition scoring using body shape signature and fourier descriptors. J. Dairy Sci. 96 (12), 8047–8059.

Bewley, J., Schutz, M., 2008. An interdisciplinary review of body condition scoring for dairy cattle. Profess. Animal Sci. 24 (6), 507–529.

Bewley, J., Peacock, A., Lewis, O., Boyce, R., Roberts, D., Coffey, M., Kenyon, S., Schutz, M., 2008. Potential for estimation of body condition scores in dairy cattle from digital images. J. Dairy Sci. 91 (9), 3439–3453.

Canny, J., 1986. A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8 (6), 679–698.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. Smote: synthetic minority over-sampling technique. J. Artificial Intell. Res. 16, 321–357.

Chetlur, S., Woolley, C., Vandermersch, P., Cohen, J., Tran, J., Catanzaro, B., Shelhamer, E., 2014. cudnn: Efficient primitives for deep learning, CoRR abs/1410.0759. <http://arxiv.org/abs/1410.0759>.

Chollet, F., et al., 2015. Keras, <https://github.com/fchollet/keras>.

Deng, L., Yu, D., et al., 2014. Deep learning: methods and applications. Found. Trends® Signal Process. 7 (3–4), 197–387.

Ding, L., Goshtasby, A., 2001. On the canny edge detector. Pattern Recogn. 34 (3), 721–725.

Fankhauser, P., Bloesch, M., Rodriguez, D., Kaestner, R., Hutter, M., Siegwart, R., 2015. Kinect v2 for mobile robot navigation: Evaluation and modeling. In: 2015 International Conference on Advanced Robotics (ICAR). IEEE, pp. 388–394.

Ferguson, J.D., Galligan, D.T., Thomsen, N., 1994. Principal descriptors of body condition score in holstein cows. J. Dairy Sci. 77 (9), 2695–2703.

Ferguson, J., Azzaro, G., Licitra, G., 2006. Body condition assessment using digital images. J. Dairy Sci. 89 (10), 3833–3841.

Fischer, A., Luginbühl, T., Delattre, L., Delouard, J., Faverdin, P., 2015. Rear shape in 3 dimensions summarized by principal component analysis is a good predictor of body condition score in holstein dairy cows. J. Dairy Sci. 98 (7), 4465–4476.

Halachmi, I., Klopčič, M., Polak, P., Roberts, D., Bewley, J., 2013. Automatic assessment of dairy cattle body condition score using thermal imaging. Comput. Electron. Agric. 99, 35–40.

Han, J., Pei, J., Kamber, M., 2011. Data Mining: Concepts and Techniques. Elsevier.

Hansen, M., Smith, M., Smith, L., Hales, I., Forbes, D., 2015. Non-intrusive automated measurement of dairy cow body condition using 3d video. In: Proceedings of the Machine Vision of Animals and their Behaviour (MVAB). BMVA Press, pp. 1.1–1.8. https://doi.org/10.5244/C.29.MVAB.1.

Heinrichs, A., Jones, C., Ishler, V., 2017. Body Condition Scoring as a Tool for Dairy Herd Management. Penn State College of Agricultural Sciences.

Hijazi, S., Kumar, R., Rowen, C., 2015. Using convolutional neural networks for image recognition.

Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K., 2016. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and < 0.5 mb model size, arXiv preprint <arXiv:1602.07360>.

Kellogg, W., 2010. Body Condition Scoring With Dairy Cattle. University of Arkansas, Division of Agriculture.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105.

Krukowski, M., 2009. Automatic determination of body condition score of dairy cows from 3d images (Master's thesis). Royal Institute of Technology, School of Computer Science and Communication.

Lachat, E., Macher, H., Mittet, M., Landes, T., Grussenmeyer, P., 2015. First experiences with kinect v2 sensor for close range 3d modelling. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. 40 (5), 93.

Maimon, O., Rokach, L., 2010. Data Mining and Knowledge Discovery Handbook, 2nd ed. Springer Publishing Company, Incorporated.

Marinello, F., Pezzuolo, A., Cillis, D., Gasparini, F., Sartori, L., 2015. Application of kinect-sensor for three-dimensional body measurements of cows. In: Proceedings of 7th European Conference on Precision Livestock Farming, ECPLF, pp. 15–18.

Markusfeld, O., Galon, N., Ezra, E., 1997. Body condition score, health, yield and fertility in diary cows. Vet. Rec. 141 (3), 67–72.

Roche, J.R., Friggens, N.C., Kay, J.K., Fisher, M.W., Stafford, K.J., Berry, D.P., 2009. Invited review: body condition score and its association with dairy cow productivity, health, and welfare. J. Dairy Sci. 92 (12), 5769–5801.

Rodríguez Alvarez, J., Arroqui, M., Mangudo, P., Toloza, J., Jatip, D., Rodríguez, J., Zunino, A., Machado, C., Mateos, C., 2017. Body condition estimation on cows from 3D images using convolutional neural networks. In: Proceedings of the 1st International Conference on Agro Big Data and Decision Support Systems in Agriculture, pp. 41–43.

Salau, J., Haas, J., Junge, W., Bauer, U., Harms, J., Bieletzki, S., 2014. Feasibility of automated body trait determination using the sr4k time-of-flight camera in cow barns. Springer Plus 3, 225.

Shelley, A.N., 2016. Incorporating Machine Vision in Precision Dairy Farming Technologies (Ph.D. thesis). University of Kentuckyhttps://doi.org/10.13023/ETD.2016.147.

Sokolova, M., Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. Inf. Process. Manage. 45 (4), 427–437.

Spoliansky, R., Edan, Y., Parmet, Y., Halachmi, I., 2016. Development of automatic body condition scoring using a low-cost 3-dimensional kinect camera. J. Dairy Sci. 99 (9), 7714–7725.

Stehman, S.V., 1997. Selecting and interpreting measures of thematic classification accuracy. Remote Sens. Environ. 62 (1), 77–89. https://doi.org/10.1016/S0034-4257(97)00083-7. <http://www.sciencedirect.com/science/article/pii/S0034425797000837>.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9.

Theano Development Team, 2016. Theano: A Python framework for fast computation of mathematical expressions, arXiv e-prints abs/1605.02688. <http://arxiv.org/abs/1605.02688>.

Wildman, E., Jones, G., Wagner, P., Boman, R., Troutt, H., Lesch, T., 1982. A dairy cow body condition scoring system and its relationship to selected production characteristics. J. Dairy Sci. 65 (3), 495–501.