DOI: 10.5748/19CONTECSI/PSE/AGB/7013

USING MACHINE LEARNING AND IMAGE PROCESSING TECHNIQUES TO ESTIMATE THE WEIGHT OF BEEF CATTLE

Gustavo Zago Canal ; https://orcid.org/0000-0003-2113-2398
Universidade Federal de Santa Catarina

Alexandre Leopoldo Gonçalves ; https://orcid.org/0000-0002-6583-2807
Universidade Federal de Santa Catarina

Leonardo Vianna ; https://orcid.org/0000-0003-4947-0938
Universidade Federal de Santa Catarina

Antonio Carlos Sobieranski ; https://orcid.org/0000-0003-1147-1900
Universidade Federal de Santa Catarina

USING MACHINE LEARNING AND IMAGE PROCESSING TECHNIQUES TO ESTIMATE THE WEIGHT OF BEEF CATTLE

ABSTRACT
Weight is an essential feature in handling beef cattle. It is used to determine the amount of medication and monitor animal health, lactation, and growth. However, rudimentary techniques are still used to weigh animals, and as a result, weight monitoring is rarely done. Therefore, this work aims to apply computer vision and machine learning techniques in order to estimate the animals' weight in a non-intrusive way, enabling weight monitoring in large farms. This study used a dataset containing 103 images of Hereford cattle and their respective weights. The dataset was submitted to two different architectures of Convolutional Neural Networks, one sequential, and one based on the DenseNet network. The results were relevant: the sequential model reached an RMSE of 57.50 kg, a MAPE of 10.2% and an R-squared of 0.33; and the DenseNet model an RMSE of 35.67 kg, a MAPE of 6,1% and an R-squared of 0.71. Consequently, it is concluded that the proposed method has the potential to estimate the weight of beef cattle, contributing to the welfare of the animals and helping the herd monitoring by the producers.

Keywords: Convolutional Neural Networks, Image Processing, Beef Cattle Weight Estimation.

1. INTRODUCTION

Adequate animal weight measurements are essential for their health care, considering they affect factors such as lactation, growth, pregnancy, and fertility (Qiao *et al*., 2021). Consequently, monitoring their weight would be of great importance for breeders, as also beneficial to animals (Hansen *et al*., 2018).

Nowadays, animal weight is usually monitored using mechanical weight scales, which present an accurate result. However, this method requires an individual weighing of the animals; it is burdensome for breeders and stressful for animals, making the method unfeasible for use on large farms. Consequently, breeders conduct only two weighings in the breeding process and consider only small groups of animals (Kashiha et al., 2014).

A low-cost alternative process can be used in smaller farms where mechanical weight scales are uncommon due to their high cost. Instead of measuring the weight of the animals, this method estimates it using body measurements. The method consists in using tapes containing relationships between the perimeter of the animal's pectoral and weight. The tapes are wrapped around the animal's chest, and a weight value is estimated according to the animal's measurement and breed (Heinrichs *et al*., 2007).

Unfortunately, both mentioned approaches cause a disturbance of the animal routine and demand a specific workforce to perform the processes. Costa *et al*. (2019) presents a study on the impact of restrictive diets and painful procedures performed on animals, affecting their welfare. Furthermore, those techniques directly affect animal habits by separating them from others. Notwithstanding, we adopted a premise in our study that animal welfare largely depends on living with other animals and the naturalness of their routine. Another issue is that the techniques are highly manual, making frequently animal weight monitoring difficult once breeders must ensure a final quality product with a focus on profitability, in addition to considering the health and welfare of the animals (Berckmans, 2014). It results in large farms conducting only two weighings, one at the beginning and one at the end of the breeding process, and only on a small group of animals in the herd (Kashiha *et al*., 2014).

Computational solutions enable frequent monitoring of the animals' weight using images and machine learning to perform weight estimation. Different methods have been proposed in the recent-year literature, applying equipment and techniques variations for processing the collected data. An example of an approach was presented by Cominotte *et al*. (2020), which extracted physical features from the animals using 3D images captured by a Kinect® model 1473. These images were processed by four different types of machine learning algorithms to estimate the weight. Moreover, Dohmen *et al*. (2021) applied a different technique, using 2D-image segmentations of the animals obtained from a Mask Region-Based Convolutional Neural Network (Mask-RCNN) model. Subsequently, the segmented images were submitted to a Convolutional Neural Networks (CNN) model to estimate the weight.

In our study, we employed a variation of these cited methods. 2D images of the dorse were processed by two CNN algorithms for weight estimation: a Sequential Model used together with a DenseNet architecture. Nevertheless, different from the previous works, the image features were not used for model inputs. Instead, the entire images were processed. Therefore, we tested the application of different data augmentation techniques on the dataset to analyze the performance of models with a broader range of image variations.

2. METHOD

In this section, we present the algorithms used for weight estimation and the main concepts about their innards. We also present the dataset and the processes used in the experiments.

2.1 Artificial Intelligence

Artificial intelligence (AI) applications are increasingly broad due to their high usefulness for solving a wide range of problems. AI contains several subareas of study with different focuses. One of them is Deep Learning, the main algorithm used in this study.

The models created from Deep Learning receive this name due to a large number of layers in the network hidden layers. We used two Convolutional Neural Networks models in the experiments: a Sequential Model and a DenseNet architecture (Huang *et al*., 2017). The CNN model has some distinct layers: the input one is a Convolution layer, then a Pooling layer, and the Dense layer as the output one.

2.2 Convolution Layer

As the name suggests, the Convolution layer is the outstanding feature of Convolutional Neural Networks, where the convolution function acts. This operation involves the application of a kernel (k), also known as the convolution filter, processing the data inputted in this layer. A linear operation is applied over each input data-position through kernel processing, which performs the sum of the values. The results of this operation are known as feature maps (Yamashita *et al*., 2018) and Equation 1 demonstrates it:

$$Y_{ij} = f\left(\sum_{i=1}^{k} \sum_{j=1}^{k} (X_{ij} * K) + b\right) \quad (1)$$

where Y represents the output value of the convolution, X is the input value, K is the convolution kernel, k is the dimensions of the kernel, b is the bias, and f is the activation function.

A significant characteristic of this layer is that its dimension is reduced when the kernel is applied to the data (Zhang *et al*., 2019). This process is essential for reducing trainable parameters of the network in high-resolution data cases. However, when the convolution process is repeatedly applied, the data size can be excessively reduced, impairing the neural network performance. Applying a margin of zeros to the input data can circumvent this problem, making the result at the end of the convolution with the same dimension as the input data.

At the end of the convolution layer, an activation function is applied to the data generated by the layer. The main objective of this activation function is to introduce nonlinearity to the network to help it capture nonlinear relationships in the data. The most common activation functions are hyperbolic tangent, sigmoid, and ReLU (Rectified Linear Units) (Gu *et al*., 2018).

2.3 Pooling Layer

In most cases, the convolution layer is followed by a Pooling layer. The role of this layer in a convolutional network is to reduce the dimension of the data, allowing the model less submitted to slight distortions and reducing the algorithm parameters for training (Yamashita *et al*., 2018; Zhang *et al*., 2019).

There are different variations to the Pooling layer; the most commons are max-pooling and mean-pooling. Max-pooling aims to extract the largest value from the area where the filter is applied. Mean-pooling enables extracting the average area values whereupon the filter is applied (Tian *et al*., 2021; Albawi *et al*., 2017; Chen *et al*., 2022).

Figure 1 shows an example of a max-pooling application, with a 2x2 filter on a 4x4 feature map.

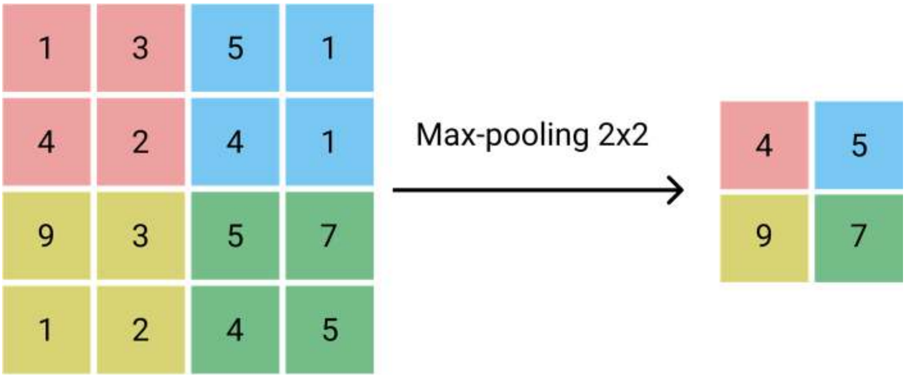


Fig. 1. Max-pooling application in a 4x4 feature map reduced to 2x2.

2.4 Dense Layer

The dense layer usually has the most significant number of connections in the network, since it is connected to all the neurons of the layers before it and to the next layer. The dense name is derived from its high-density connections.

The central negative aspect of this layer is the computational cost required to train it. Each connection can be adjusted to improve network results (Yamashita *et al*., 2018; Zhang *et al*., 2019). A technique applied to circumvent this problem is known as a dropout. It consists of randomly deactivating some neurons from the network, decreasing the number of connections necessary for training, and helping with issues such as overfitting (Albawi *et al*., 2017).

2.5 Network Models

We used two methods for estimating weight. The central architecture used was a DenseNet, but tests were performed using a Sequential Model for comparison with this DenseNet model. The TensorFlow® library was used to develop both network models, available for the Python® programming language. This library provides an API with several layer implementations and functions to develop machine learning models and complete architectures using DenseNet.

ISSN 2448-1041 - TECSI – FEA USP SÃO PAULO/ BRAZIL

a) Sequential Model

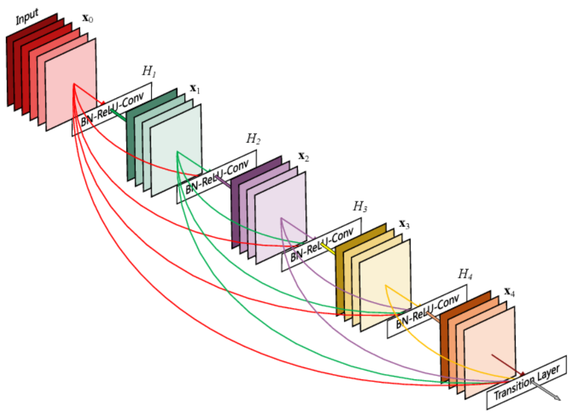
The Sequential Model implements a simple neural network architecture. It consists of three convolutional layers followed by an activation function, a BatchNormalization layer, and a max-pooling layer. As the name suggests, the model receives a data input, which is processed sequentially. Layer after layer, the data is processed and outputted to the next one. The model repeats this procedure until reaching the output layer and, finally, providing a result for weight estimation.

b) DenseNet Model

The network model that applies DenseNet's architecture is more complex. This architecture was proposed in 2017; it has a different process from the conventional one and is composed of dense blocks.

These dense blocks have convolutional layers. However, the feature maps generated by these layers are outputted to all subsequent convolution layers within the dense block, generating a network connection number greater than a conventional neural network. Equation 2 describes the number of connections C of a dense block in length L.

$$C = \frac{L\,(L + 1)}{2} \tag{2}$$



For a clearer vision of how images are processed in a DenseNet, consider an image x_0, loaded in a convolutional network with a layer L. Each layer uses a nonlinear transformation $H_l(.)$, where l is the layer index. Operations executed by BatchNormalization (BN), ReLU, and Pooling layers represent this nonlinear transformation. According to Huang *et al*. (2017), the direct connection of each layer to subsequent layers is implemented in the DenseNet to increase the information flow between layers. Figure 2 illustrates these connections.

Fig. 2. Dense block with five layers, illustrating that each layer receives the feature maps from all previous layers.

Equation 3 describes the process illustrated by Figure 2 for one layer l.

$$x_l = H_l([x_0, x_1, \ldots, x_{l-1}]) \tag{3}$$

where x_l is the feature maps of layer l and $[x_0, x_1, ..., x_{l-1}]$ represents the feature maps concatenation of the previous layers.

Originally, DenseNet was trained with the ImageNet dataset, which consisted of one thousand classes for classification models and is widely used as a benchmark for neural networks. This network had four dense blocks, with a BatchNormalization layer connecting them, and a 1x1 kernel convolution layer followed by a 2x2 mean-pooling layer.

Because DenseNet is a classification network and our work settled a regression problem, some adjustments were necessary to fit this purpose. Therefore, we removed the 7x7 mean-pooling layer and the dense output layer of one thousand nodes positioned at the end of the network. In their places, a 2D global average pooling layer and just one dense node (as the output layer) were added because the expected output was a single value for the animal's weight.

2.6 Dataset

The dataset used in our research was obtained from the study conducted by Ruchay *et al*. (2020). It consists of 103 images of the top and side views of Hereford breeds from a Russian farm. Ten features had been manually extracted from these images, including withers height, hip height, chest depth, heart girth, ilium width, hip joint width, oblique body length, hip length, and chest width.

2.7 Augmentation

The data augmentation step consisted in modifying the dataset to increase the amount of data for model training and validation. Because the dataset used in this work contains few samples, tests were performed with two different data augmentation techniques to compare the performance of both architectures.

The first test applied only a horizontal rotation to the dataset images. The mirror images were appended to the original dataset, doubling the images from 103 to 206. The second executed test included more complex procedures. Each dataset image was randomly rotated, at an angle between -54º to +54º, and was also randomly mirrored on the axis x or y. This process was performed three times for each image and resulted in 412 images when appended to the original dataset.

2.8 Model adaptations

Three convolutional layers were used to implement the Sequential Model, each one followed by a BatchNormalization and max-pooling layer. The model had a single dense node as an output layer, implemented to solve the regression problem.

The model with the DenseNet architecture also required some modifications. As previously commented, the DenseNet was initially developed and trained with the ImageNet dataset of one thousand classes. For this reason, some modifications to the structure of the network were necessary for a regression output.

2.9 Training and Evaluation

The training process was performed after adapting the network models. In this step, the connection weights between the network layers were repeatedly recalculated to decrease their prediction error. For this step, the dataset was separated into two parts, one for network training and the other for testing the final models.

The dataset was split into 90% for training and 10% for testing. From the training dataset, 80% were used for model training and 20% for the validation process performed subsequently to training. The DenseNet model training process executed 50 epochs to converge. Ten thousand epochs were executed for the Sequential Model due to its simpler architecture.

2.10 Metrics for evaluating results

Three metrics were applied to measure the performance of the models: coefficient of determination (R^2), root mean squared error (RMSE), and mean absolute percentage error (MAPE). Table I exhibits the equation to calculate each one, where y_i represents the actual value of the animal's weight for a given image, $\hat{y}_i$ represents the estimated value, y is the average weight of the test dataset, and n is the test dataset size.

The metrics were selected to ensure that each one would present the model error in a different format. R^2 represents the ratio between the unexplained variation by the total variation of the model predictions. RMSE represents a value in kilograms for the model errors and MAPE is an error value in percentage.

Table I. Metrics used for model evaluation.

Notation	Metric	Formula	
R^2	Coefficient of determination	$1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \underline{y})^2}$	(4)
RMSE	Root mean squared error	$\sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$	(5)
MAPE	Mean absolute percentage error	$\frac{1}{n}\sum_{i=1}^{n} \frac{\lvert y_i - \hat{y}_i \rvert}{\lvert y_i \rvert} * 100$	(6)

3. RESULTS AND DISCUSSION

We performed six experiments with different data augmentation techniques applied to the dataset. A model was trained for each architecture and each dataset variation, using the set without modifications, with horizontal mirroring, and with random mirroring and rotations.

Table I shows the model results for different data augmentation. Analyzing the outcomes, the DenseNet model outperformed the Sequential Model, with better results in all scenarios, excluding test 3, involving mirroring and rotations in the dataset images.

3.1 Sequential Model

As seen in Figure 3, the Sequential Model obtained significantly better performance in the first test. However, the model had a strange behavior for animals weighing more than 475 kg in tests 2 and 3, yielding gross errors that were penalized by the metrics. The most severe errors produced by tests 2 and 3 emerged in predicted values lower than the actual animals' weight and not higher.

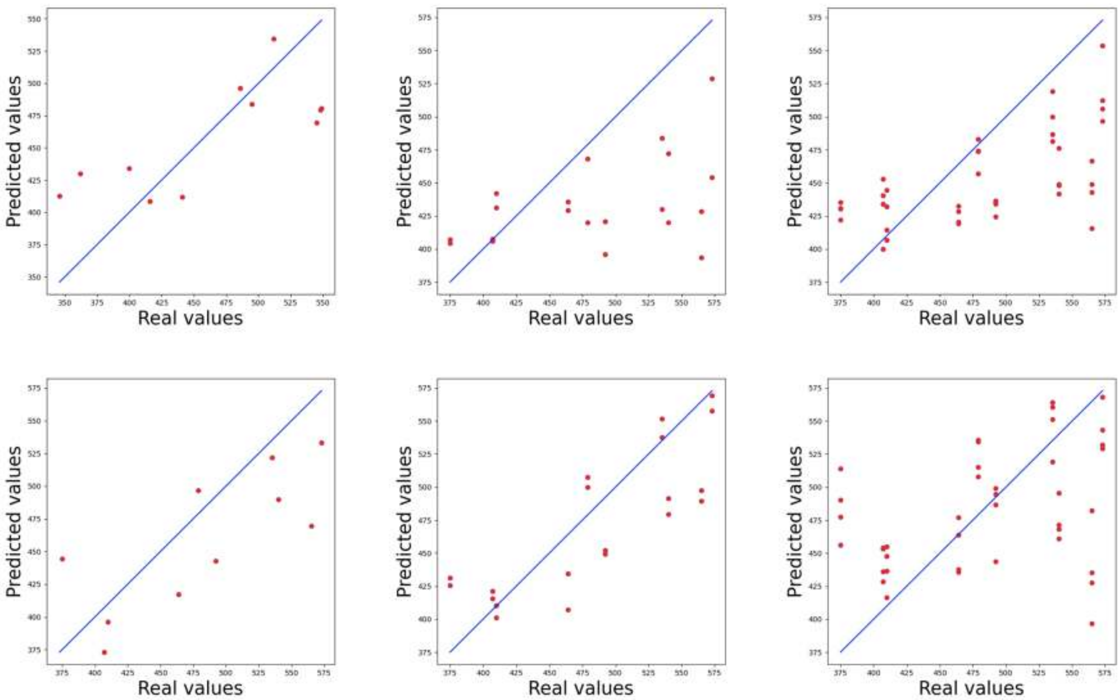


Fig. 3 Comparison graphs of the real values and the values predicted by each model.

3.2 DenseNet Model

The DenseNet model presented remarkable results for weight prediction. When performing the analysis of Figure 3, we observed better results in the first two tests, as also demonstrated by Table I.

In the first two tests, the model produced estimations close to the actual values without serious errors that could impact the metrics. Inversely, the DenseNet model resulted in more severe errors in the third test, just like it occurred in the Sequential Model. However, when analyzing the predictions, the errors made were estimating much higher weights for light animals and low weights for higher-weight animals, with different behavior from the Sequential Model. The graphs presented in Figure 3 demonstrate the results of the models.

3.3 General Discussion

Table II presents the best cases of each model. The DenseNet model obtained a result of ±35.67 kg, corresponding to a percentage error of 6.1 and an R^2 of 0.71. Furthermore, the Sequential Model presented a result of 57.50 kg, with an RMSE, a MAPE of 10.2%, and an R2 of 0.33. Remarkably, the best Sequential model performance was not related to the same tests in all metrics.

To better understand this fact, it is necessary to understand how metrics penalized the error severity. Equation (5) of Table I refers to the RMSE metric, which calculates the difference between the actual value of the animal's weight (y) and its predicted value ($\hat{y}$), subsequently squaring this difference. Meanwhile, MAPE, presented in Equation (6), calculated this difference but applied its modulus and divided it by the actual value. Therefore, considering a heavier animal, the larger divisor would reduce the accumulated error value even with a large error in the dividend equation. This effect could be considered one of the reasons that the MAPE metrics presented a substantially higher value in test 3 compared to the other tests.

Likewise, R^2 results could lead to some doubts when analyzed, mainly because, in some cases, its results were negative values. The equation of the R^2 metric, demonstrated in Table I, should be interpreted to understand the reason for this behavior. The equation initially calculated the sum of the differences between the average value of the animal weights and the actual values. Therefore, the initial result was squared and divided by the sum of the differences between the actual and predicted value of the weight, which is also squared. After calculating this division, the obtained value was subtracted from one to obtain the final value for the metric. Therefore, the closer to one, the better the obtained result since the error between the actual and predicted value of the animals is low. However, in cases where the model did not present good results, the values in the dividend of the equation were higher, and the division resulted in a number superior to one. Consequently, the final subtraction generated a negative value. In cases where negative values were obtained for R^2, the predicted values differed significantly from the actual values (Figure 3).

Figure 4 illustrates the several variations of the images in the experiments, the respective errors, and predicted values obtained from the models, allowing the understanding of performed tests. The part identified by (a) refers to the entire dataset. Part (b) presents an example of the dataset with horizontal mirroring, and part (c) demonstrates the test with mirroring and rotation procedures.

Figure 4 shows that the Sequential Model performed better than the DenseNet in most cases. Because a low amount of data was considered to support this analysis, it cannot be considered definitive. The Sequential Model produced excellent results; however, it generated a significant error in some cases, affecting the overall result of its evaluation. Likewise, part (c) of Figure 4 presents an example of when the DenseNet model produced an error greater than 10% and, when applied, the RMSE metric significantly impacted the overall evaluation of the results. The R^2 metric was not applied in this comparison because it used average values in this calculation; and, in this case, the average value would be only one for each test and could not represent an unbiased value for this metric.

Fig. 4 Comparison between values predicted by the two models in the three tests performed with the negative dataset.

Table II. Results of the executed experiments, applying different strategies for data

(1)

Image	Model	Real value	Estimated value	Error (kg)	Error (%)
1	Sequential	410	403	7	1,71
1	DenseNet	410	396	14	3,41

(1) (2)

Image	Model	Real value	Estimated value	Error (kg)	Error (%)
1	Sequential	410	442	32	7,8
1	DenseNet	410	447	37	9,02
2	Sequential	410	431	21	5,12
2	DenseNet	410	428	18	4,39

(1) (2) (3)

Image	Model	Real value	Estimated value	Error (kg)	Error (%)
1	Sequential	410	444	34	8,29
1	DenseNet	410	447	37	9,02
2	Sequential	410	414	4	0,98
2	DenseNet	410	455	45	10,98
3	Sequential	410	407	3	0,73
3	DenseNet	410	416	6	1,46

augmentation of the dataset.

Dataset	Metric	Sequential Model	DenseNet Model
Without augmentation	RMSE	**±57.50 kg**	±49.41 kg
Horizontal mirroring	RMSE	±77.50 kg	**±35.67 kg**
Mirroring and rotation	RMSE	±61.34 kg	±61.73 kg
Without augmentation	MAPE	10.8%	8.9%
Horizontal mirroring	MAPE	11.9%	**6.1%**
Mirroring and rotation	MAPE	**10.2%**	10.0%
Without augmentation	R^2	**0.33**	0.45
Horizontal mirroring	R^2	-0.37	**0.71**
Mirroring and rotation	R^2	-0.02	0.13

4. FINAL REMARKS

Monitoring animals' weight is an essential aspect of beef cattle raising. The present study addressed the use of computational techniques involving machine learning and image processing to estimate the weight, enabling a less costly animal monitoring process for breeders, which could be applied more frequently.

Two machine learning algorithms were presented for this task: one was a Sequential Model, and the other applied the DenseNet architecture. The models obtained significant results in the performed tests. In its best performance, the Sequential Model resulted in ±57.50 kg of RMSE, a MAPE of 10.2%, and an R^2 of 0.33. The model with the DenseNet architecture stood out by obtaining ±35.67 kg of RMSE, MAPE of 6.1%, and R^2 of 0.71.

In general, the results obtained in this study were promising. They could be optimized if a dataset with more angle variations and a more extensive data volume were used for training the models, leading to better results.

ACKNOWLEDGMENT

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

REFERENCES

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. In 2017 international conference on engineering and technology (icet) (pp. 1–6).
Berckmans, D. (2014, 04). Precision livestock farming technologies for welfare management in intensive livestock systems. Revue scientifique et technique (International Office of Epizootics), 33, 189-96. doi: 10.20506/rst.33.1.2273
Chen, J., Wang, S., He, E., Wang, H., & Wang, L. (2022). Two-dimensional phase lag index image representation of electroencephalography for automated recognition of driver fatigue using convolutional neural network. Expert Systems with Applications, 191, 116339.

Retrieved from https://www.sciencedirect.com/science/article/pii/S0957417421016377 doi: https://doi.org/10.1016/j.eswa.2021.116339

Cominotte, A., Fernandes, A., Dorea, J., Rosa, G., Ladeira, M., van Cleef, E., Pereira, G., Baldassini, W., Machado Neto, O. (2020). Automated computer vision system to predict body weight and average daily gain in beef cattle during growing and finishing phases. Livestock Science, 232, 103904. Retrieved from https://www.sciencedirect.com/science/article/pii/S1871141319310856 doi: https://doi.org/10.1016/j.livsci.2019.103904

Costa, J. H., Cantor, M. C., Adderley, N. A., & Neave, H. W. (2019). Key animal welfare issues in commercially raised dairy calves: social environment, nutrition, and painful procedures. Canadian Journal of Animal Science, 99(4), 649-660. Retrieved from https://doi.org/10.1139/cjas-2019-0031 doi: 10.1139/cjas-2019-0031

Dohmen, R., Catal, C., & Liu, Q. (2021). Image-based body mass prediction of heifers using deep neural networks. Biosystems engineering, 204, 283–293.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., Chen, T. (2018). Recent advances in convolutional neural networks. Pattern Recognition, 77, 354-377. Retrieved from https://www.sciencedirect.com/science/article/pii/S0031320317304120 doi: https://doi.org/10.1016/j.patcog.2017.10.013

Hansen, M., Smith, M., Smith, L., Abdul Jabbar, K., & Forbes, D. (2018). Automated monitoring of dairy cow body condition, mobility and weight using a single 3d video capture device. Computers in Industry, 98, 14-22. Retrieved from https://www.sciencedirect.com/science/article/pii/S016636151730475X doi: https://doi.org/10.1016/j.compind.2018.02.011

Heinrichs, A., Erb, H., Rogers, G., Cooper, J., & Jones, C. (2007). Variability in holstein heifer heart-girth measurements and comparison of prediction equations for live weight. Preventive Veterinary Medicine, 78(3), 333-338. Retrieved from https://www.sciencedirect.com/science/article/pii/S0167587706002455 doi: https://doi.org/10.1016/j.prevetmed.2006.11.002

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the ieee conference on computer vision and pattern recognition (pp. 4700–4708).

Kashiha, M., Bahr, C., Ott, S., Moons, C. P., Niewold, T. A., Ödberg, F. O., & Berckmans, D. (2014). Automatic weight estimation of individual pigs using image analysis. Computers and Electronics in Agriculture, 107, 38-44. Retrieved from https://www.sciencedirect.com/science/article/pii/S0168169914001525 doi: https://doi.org/10.1016/j.compag.2014.06.003

Qiao, Y., Kong, H., Clark, C., Lomax, S., Su, D., Eiffert, S., & Sukkarieh, S. (2021). Intelligent perception for cattle monitoring: A review for cattle identification, body condition score evaluation, and weight estimation. Computers and Electronics in Agriculture, 185, 106143. Retrieved from https://www.sciencedirect.com/science/article/pii/S0168169921001617 doi: https://doi.org/10.1016/j.compag.2021.106143

Ruchay, A., Kober, V., Dorofeev, K., Kolpakov, V., & Miroshnikov, S. (2020). Accurate body measurement of live cattle using three depth cameras and non-rigid 3-d shape recovery. Computers and Electronics in Agriculture, 179, 105821. Retrieved from https://www.sciencedirect.com/science/article/pii/S0168169920321256 doi: https://doi.org/10.1016/j.compag.2020.105821

Tian, J., Geng, F., Zhao, F., Gao, F., & Niu, X. (2021, July). Application of convolutional neural network in fault line selection of distribution network. Journal of Applied Science and Engineering, 25, 195–205. Retrieved from https://doi.org/10.6180/jase.202202_25(1).0020 doi: 10.6180/jase.202202_25(1).0020

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. Insights into imaging, 9(4), 611–629.

Zhang, Q., Zhang, M., Chen, T., Sun, Z., Ma, Y., & Yu, B. (2019). Recent advances in convolutional neural network acceleration. Neurocomputing, 323, 37-51. Retrieved from https://www.sciencedirect.com/science/article/pii/S0925231218311007 doi: https://doi.org/10.1016/j.neucom.2018.09.038