

# Cattle Breed Identification and Live Weight Evaluation on the Basis of Machine Learning and Computer Vision

Oleg Rudenko<sup>1</sup>[0000-0003-0859-2015], Yuriy Mege<sup>2</sup>[0000-0003-0072-0756],  
Oleksandr Bezsonov<sup>1</sup>[0000-0001-6104-4275], Antonina Rybalka<sup>1</sup>[0000-0002-4148-2443]

<sup>1</sup> Kharkiv National University of Radio Electronics, Nauky Ave. 14, Kharkiv, 61166,  
Ukraine

oleh.rudenko@nure.ua  
oleksandr.bezsonov@nure.ua  
antonina.rybalka@nure.ua

<sup>2</sup> Kharkov National Technical University of Agriculture named after P. Vasilenko,  
Alchevskih str., 44, Kharkiv, 61002, Ukraine  
megelye@gmail.com

**Abstract.** The problem of the cow's live weight estimation is considered. A convolutional neural network based method for animal recognition and its breed identification in combination with epipolar geometry approach for object's size measurement is proposed. Information regarding animal's size and its breed is further used for LW estimation by multilayer perceptron based predictive model. This approach can be used to replace traditional methods of direct observation and measurement. The proposed system can be widely used in the management of a modern farm. Accuracy and performance of the proposed method has been tested with the participation of the experts.

**Keywords.** Convolutional neural network, epipolar geometry, image processing, computer vision, cow, live weight, mask-rcnn

## 1 Introduction

Computer-based image analysis and various predictive applications are commonly used in different fields of human activity, one of which is agriculture. The number of farms in Ukraine has increased significantly and their productivity is growing, so the importance of computer technology in the automation of agricultural processes is

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

gradually increasing. When raising cows, the relationship between live weight (LW), milk yield and feed intake can be taken as criteria for organizing the care and nutrition of animals in modern keeping conditions. These parameters are quite important and must be strictly controlled. When they go beyond permissible limits, this significantly affects the immune system of cows, and, accordingly, the economic efficiency of the farm. Negative changes in live weight may indicate animal's health problems, inappropriate environmental conditions, and nutritional errors. Therefore, a parameter such as live weight (LW) is certainly important for dairy cows [1]. It should be noted that at present the process of measuring and servicing cattle is still carried out manually and is very expensive.

## **2 Formal problem statement**

The aim of this study is the estimation of cows LW using several neural network models: the convolution artificial neural network for recognition of a cow at the pictures and its breed identification with subsequent determination of the size of its body by the stereopsis method with subsequent utilization of multilayer perceptron for LW estimation of a cow on the basis of information regarding its breed and size.

For more accurate estimation of the animal's physical parameters 3D camera (Intel RealSense D435i) was additionally used. It should be noted that the use of a 3D camera alone does not yield to good results due to its low resolution. Thus, cows images taken at different angles are used to determine parameters of bodies cows using photogrammetric method. Parameters such as the withers height (WH), hip height (HH), the body length (BL) and the hip width (HW) of cows were obtained via photogrammetry.

Model estimation based on the ANN has been developed using those parameters (input parameters WH, HH, BL, HW, and the output parameter - LW).

Cow's body dimensions are determined from the analysis of animal images taken synchronized cameras from different perspectives. Initially, a cow is identified at the image and its breed is determined by using Mask-rcnn convolution neural network. Then withers height, hip height, length and width of a cow determined via stereopsis method which allows to obtain geometric parameters of the objects at digital images and perform their measurements. Digital imaging and photogrammetric processing include several completely certain steps that can generate three-dimensional or two-dimensional digital model of the animal's body. Then obtained data about the species and its size are fed to the predictive model, which determines the estimated weight of the animal.

## **3 Literature review**

In the literature different estimation of animals LW for various purposes is typically performed by using regression equations. However, in recent years the methods and means of computational intelligence based on ANN are increasingly used.

Prediction of LW of bulls' slaughter value from growth data by using ANN carried out in [2], an artificial intelligence technology in dairy industry [3], a comparison of an artificial neural network and a method of linear regression for prediction of the LW of hair goats [4], comparison of ANN and decision tree algorithms used for prediction of LW at post weaning period [5], weight prediction of broiler chickens with the help of 3D-computer vision is performed in [6], prediction of the goats masses [7], artificial neural network to predict rabbits body weight [8], prediction of carcass meat percentage in young pigs using linear regression models and artificial neural networks [9], weighing pigs using machine vision and artificial neural networks [10].

The purpose of this study is to estimate cows LW by using artificial neural networks. Body dimensions (BD) were determined by the photogrammetric method on images in which the cow is identified and classified by the convolution neural network.

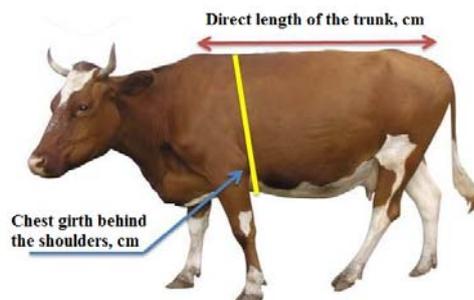
## 4 The traditional methods of measuring the mass of cows

### 4.1 The method of Trukhanovskii

This method is used to determine LW of the adult cattle by the following formula

$$LW = \frac{A \cdot B}{100} \cdot K, \quad (1)$$

where  $A$  – chest girth behind the shoulders, cm;  $B$  - direct length of the trunk, measured with a stick, cm;  $K$  – a correction factor (2 - for dairy cattle and 2.5 - for milk-meat and beef breeds).



**Fig. 1.** The measurement method of Trukhanovskii

To determine the approximate body weight special tables are used. For these tables the initial data is measurements of animals taken at the correct animal's position (feet must stand upright, a head at the level of the back).

## 4.2 Other methods

There are some other methods of LW and cow's body size estimation that could be applied, in particular Pin Bone method [11]

$$BW = \frac{((HG)^2 \times BL)}{300}, \quad (2)$$

where BW – Body Weight estimation (pounds); HG - heart girth (inch); BL - Body Length (inch).

Thus, in estimating of cattle LW the important task of determining its size measure arises. This problem is complicated especially when measures are made in real conditions. The traditional animal measuring process is shown in Fig. 2.



Fig. 2. The traditional measuring process

## 5 Determination of the size of the cattle using the stereopsis method

The first step in the proposed algorithm is detection of a cow.

This step is based on the use of convolution neural network for the cows detection at the picture and stereopsis method, which allows the system to obtain measurements of the real world objects, located at different distances from the cameras.

### 5.1 The stereopsis principle

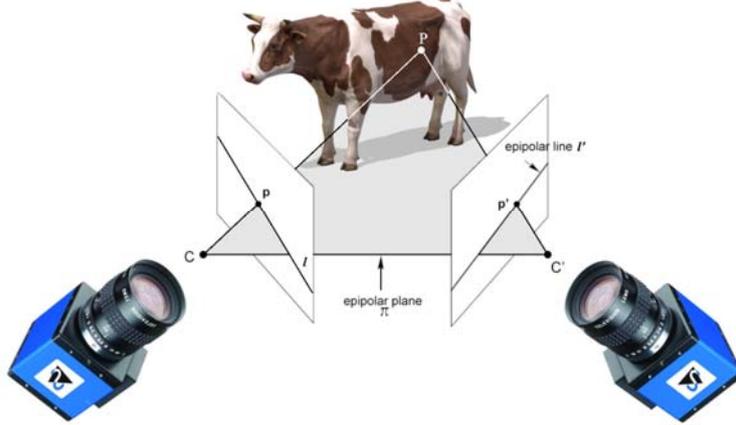
This principle can be explained as follows:

Suppose there are two cameras, defined by their matrices  $P$  and  $P'$  in some coordinate system. In this case we say that there is a pair of calibrated cameras. If the cameras centers do not match, the pair of cameras can be used to determine the three-dimensional coordinates of the observed points.

Often, the coordinate system is chosen in such a way that the cameras matrix are of the form  $P = K[I | 0]$ ,  $P' = K'[R' | t']$  (this is always possible, if we choose the origin coinciding with the first camera's center, and direct a Z-axis along the optical axis).

Consider a point  $P$  in the real three-dimensional space projected simultaneously in two image points  $p$  and  $p'$  through the two camera projection center ( $C$  and  $C'$ ). The points  $P$ ,  $p$ ,  $p'$ ,  $C$  and  $C'$  lie in a plane, called the epipolar plane. Epipolar plane inter-

sects each image forming the intersection lines. These lines ( $l$  and  $l'$ ) correspond to the projection ray through the  $p$  and  $P$ , and  $p'$  and  $P$ , and are called epipolar lines. This projection epipolar geometry is described in [12] (Fig. 3).



**Fig. 3.** Epipolar geometry

Epipolar geometry is used to search for stereo pairs and for verifying that a pair of points can be a stereo pair (i.e. the projection of a point in space).

Epipolar geometry has a very simple description in the coordinates. Suppose there is a pair of calibrated cameras, and let  $p$  is homogeneous coordinates of the point at the image from the first camera, and  $p'$  - from the second camera. A  $3 \times 3$  matrix  $F$  exists, such that the pair of points  $p, p'$  is a stereo pair if and only if:

$$p'^T F p = 0. \quad (3)$$

The matrix  $F$  is called a fundamental matrix. Its rank is 2 and it is determined up to a nonzero factor that only depends on the original matrix cameras  $P$  and  $P'$ .

In the case where the matrix cameras are of the following form  $P = K[I | 0]$ ,  $P' = K'[R | t]$  the fundamental matrix may be calculated with the formula:

$$F = (K'^{-1})^T R K^T [K R^T t]_x, \quad (4)$$

where the  $e$  vector notation  $[e]_x$  calculated as

$$[e]_x = \begin{bmatrix} 0 & -e_z & e_y \\ e_z & 0 & -e_x \\ -e_y & e_x & 0 \end{bmatrix}. \quad (5)$$

In [13] the various calculation algorithms of  $F$  with using a set of points are considered. In particular, gradient descent algorithm, Newton's method and the Levenberg-Marquardt algorithm are described.

Epipolar line equations are calculated with the help of the fundamental matrix. For the point  $x$ , the vector that defines the epipolar line is of the form  $l' = Fp$ , and the equation of the epipolar line itself is:  $l'^T p' = 0$ . Similarly, for a point  $p'$ , the vector defining the epipolar line is of the form  $l = F^T p'$ .

## 5.2 Construction of the depth map

Depth map - is an image in which each pixel, instead of color, keeps a distance from the camera. To this end, for each point in one image its pair is searched on the other image. A pair of corresponding points can be used for triangulation and determination of their prototype coordinates in three dimensions. Knowing the three-dimensional coordinates of the prototype image, the depth is calculated as the distance to the camera's plane [13].

Thus it is possible to determine the size of the object.

## 6 Convolutional Neural Network (CNN)

CNN was first proposed in [14] as the development of the neocognitron model [15], intended for effective image recognition. This network uses pattern recognition technology based not on hardcoded by developers algorithms, but on system training, which involves a consistent identification of a huge number of images.

Subsequently, the R-CNN (Regions With CNNs) networks were built based on CNN. R-CNN were used for detecting all objects of the given classes and determining the bounding box for each of them (object detection). R-CNN creates bounding boxes for each object in the image or suggestions regions using selective search process. Fast R-CNN increases productivity of R-CNN and classifies objects of each region together with tighter bounding boxes. Next network, Faster R-CNN, improved generation mechanism of used therein candidate regions by computing the regions not on the original image but on the features map derived from CNN. For this purpose the module called Region Proposal Network (RPN) was added.

Finally, the Mask R-CNN network [16] improves Faster R-CNN architecture by adding one more sub-module, which predicts the position of the detected object covering mask, and thus solves the task of instance segmentation. After image processing, the network outputs objects bounding boxes (bbox), their classes (class) and masks (mask). It worth to note, that Mask R-CNN is one of the fastest network at the moment. The structure of this network is shown in Fig. 4.

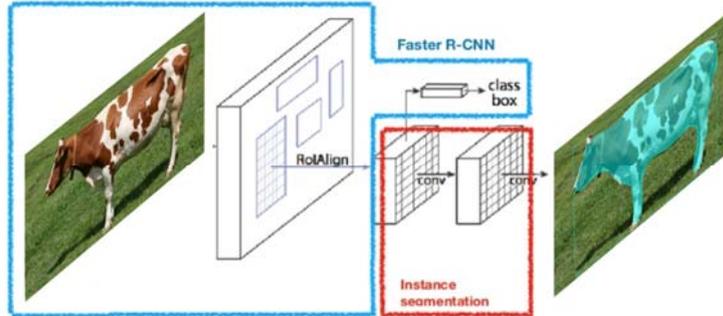


Fig. 4. Mask R-CNN structure

### 6.1 The architecture of the neural network convolution

Initially, the convolution neural network structure was created taking into account the structural features of some parts of the human brain responsible for vision. Three mechanisms are laid into the foundation for the development of such networks:

- local perception;
- building a set of layers in the form of characteristics maps (shared weights);
- sub-sampling.

In accordance with these arrangements, three main layers are used to build a convolutional neural network: convolution, pooling (otherwise subsampling or down-sampling), fully connected layer.

### 6.2 Convolution layer

Convolution equation for the  $l$ -th ( $l=1, \dots, L$ ) network layer has the following form [17]

$$x_{ij}^l = \sum_{a=-\infty}^{+\infty} \sum_{b=-\infty}^{+\infty} w_{ab}^l \cdot y_{(is-a)(js-b)}^{l-1} + b^l. \quad (6)$$

It reflects movement of the  $w^l$  core along the image or map of input attributes for the layer  $y^{l-1}$ . Here  $y^{l-1} = f(x_{ij}^{l-1})$  - image after  $(l-1)$ -th layer;  $f(\bullet)$  - used activation function;  $b^l$  - offset. Indices  $i, j, a, b$  - indices of the elements in the matrices,  $S$  - the value of the convolution step.

As seen from (3) convolution are performed for each element  $i, j$  of image matrix  $x^l$ .

Convolution preserves spatial relationships between the pixels.

Each convolutional layer is followed by subsampling or computational layer that is serving to reduce the dimension of the image by averaging the values of the local output neurons.

### 6.3 Subsampling Layer (pooling, MAX-pooling)

Subsampling layer zooms planes by local averaging of the output neurons values . Thus, the hierarchical organization is achieved. Subsequent layers are extracted more common characteristics that less depend on the image distortion.

The difference between a subsampling layer and a convolution layer is that in the convolution layer regions of the neighboring neurons overlap, which does not occur in the subsampling layer.

Pooling layer operates independently of the input data depth and scales the spatial volume by using a maximum function.

The architecture of the convolution network assumes that the presence of a sign is more important than information about its location. Therefore, the maximum one is selected from several neighboring neurons in the feature map and its value is considered as a single neuron in the feature map of lower dimension.

In addition to maximum subsampling, pooling layers can perform other functions, such as averaging subsampling or even L2-normalized subsampling.

### 6.4 Non-linear activation layer

On these layers within the network, a nonlinear activation function  $f(\bullet)$  is applied to all input values and the result is sent to the output. Thus, the activation layer does not change the input resolution.

Usually due to significant positive properties for hidden layers, ReLu(ReLU (x) = max (0; x)) and its various modifications (Leaky ReLU, Parametric ReLU, Random-

ized ReLU) [18] are used. SoftMax function  $f_j^L = e^{x_j^L} \left( \sum_{i=1}^{N^L} e^{x_i^L} \right)^{-1}$  (for solving clas-

sification problems) or linear function (for regression tasks) are used for fully connected layer.

### 6.5 Dropout layer

Different regularization techniques are used to avoid network retraining.

Dropout is a simple and effective regularization method and consists in the fact that in the process of training a network, a subnet is randomly allocated from its aggregate topology, i.e. part of the neurons is turned off from the process, and the next update of the scales occurs only within the allocated subnet. Thus, only weights of remaining neurons are changed. Each neuron is excluded from the total network with a certain probability, which is called the dropout rate.

This layer reduces the time of one training epoch due to the smaller number of optimized parameters, and also allows to better deal with retraining of the network compared to standard regularization methods.

## 6.6 Normalizing layer

The standard normalization of inputs occurs on this layer (the sample average of their values is subtracted, and the result is divided by the root of the sample variance). Sampled values are calculated taking into account the values at the inputs of this layer at previous training iterations. This approach allows to increase the speed of learning the network and improve the final result.

## 6.7 Fully connected layer

This layer is an ordinary multilayer perceptron, the purpose of which is classification. It models a complex nonlinear function, the optimization of which improves the quality of recognition.

The neurons of each map of the previous subsample layer are associated with one neuron of the hidden layer. Thus, the number of neurons in the hidden layer is equal to the number of maps in the subsample layer.

As in ordinary neural networks, in a fully connected layer, neurons are connected to all activations in the previous layer. Their activations can be calculated by multiplying matrices and applying bias.

The differences between fully connected and convolutional layers are that the neurons of the convolutional layer

- 1) connected only with the local area of the input;
- 2) can share parameters.

## 6.8 The choice of training criteria

ANN training is an iterative process. At each iteration, the network outputs for one (or more) samples in the training set are calculated, and the network weights are adjusted to reduce the error between the actual network output  $(y_{i,p}^L)(i=1,\dots,N^L)$  and target output for a given sample  $d_{i,p}$ . Therefore, training is reduced to minimizing some error function.

In practice, a quadratic function, cross entropy, or some combined functional are used as a criterion for the error function.

## 7 Backpropagation Neural Network (BPNN)

Backpropagation neural network (BPNN) is a multilayer feedforward neural network that uses a supervised learning algorithm known as error back-propagation algorithm. Errors accumulated at the output layer are propagated back into the network for the adjustment of weights.

Multilayer perceptron (MLP) is a neural network with several layers, each of which consists of computing nodes (neurons). The topology of the MLP is shown in Fig. 5 [19]. Network inputs are connected to each neuron in the first layer. The outputs of

the neurons of the first layer then become inputs to the neurons of the second layer and so on. The last layer is the output layer, all other layers between the input and output layers are called hidden layers. The architecture of the multi-layer perceptron can be conveniently written as  $n_0 - n_1 - \dots - n_l$  where  $n_0$  is the network's input vector dimension, and  $n_i$ ,  $1 \leq i \leq l$  denotes the number of nodes in the respective layers.

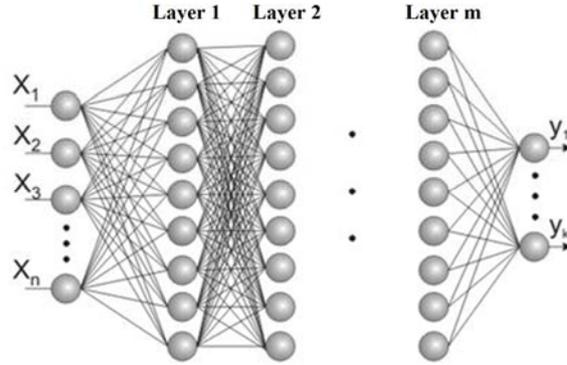


Fig. 5. Backpropagation neural network

Thus, MLP uses the following approximation of the nonlinear operator  $f(\bullet)$ :

$$\hat{y}(k) = \hat{f}(k) = f^q \left[ \left( W^q \right)^T f^{q-1} \left[ \left( W^{q-1} \right)^T f^{q-2} \left[ \dots f^1 \left[ \left( W^1 x(k) + b_1 \right)^T \right] \dots \right] \right] + b_q, \quad (7)$$

where  $W^i$  - vector of weight parameters of neurons in the  $i$ -th layer of the network;  $f^i[\bullet]$  - activation function (AF) of  $i$ -layer,  $b_i$  - bias of the  $i$ -th neuron.

Since in practical applications it is necessary to perform various operations not only with the activation function itself, but also with its first derivative, it is necessary to use a monotonous, differentiable and limited function as an activation function. A particularly important role is played by such functions in modeling nonlinear relationships between input and output variables. These are the so-called logistic or sigmoidal (S-shaped) functions.

In general, the representation of the input-output can be expressed as  $\hat{f}: R^{n_l} \rightarrow R^m$ .

Approximate capacity of MLP have been studied and described by many authors [19-23]. The basic idea is that every continuous function  $f: D_f \subset R^{n_l} \rightarrow R^m$  can be uniformly approximated with arbitrary precision function  $\hat{f}$  by  $D_f$ , where  $D_f$  is a

compact subset of  $R^m$ , provided that there is a sufficient number of hidden layers in the network. This is true even for networks with only one hidden layer. A typical assumption for the activation functions in the hidden layer is the following:  $f(\bullet)$  is a continuous, non-constant and limited.

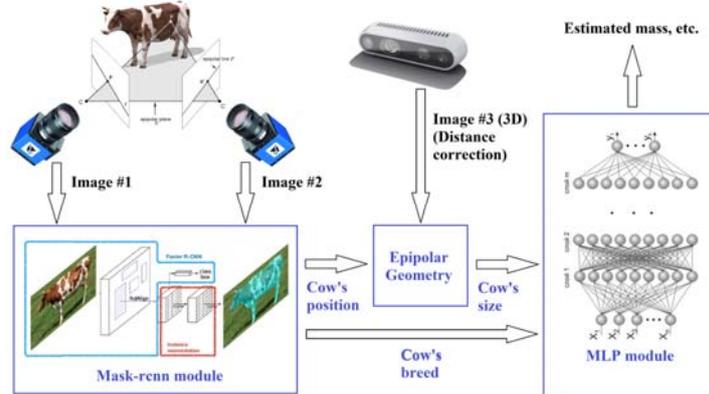
MLP is in general function approximator and it ensures that the network with one hidden layer will always be enough to represent any arbitrary continuous function. But this statement says nothing about the number of neurons in the hidden layer, which provide a given accuracy of approximation. It is also important that the proof of the possibility of approximation by means of MLP suggests that weights are set correctly. However, the question of a learning algorithm choice remains open.

The pseudocode algorithm for BPNN is given below [24].

- (I) Network initialization: randomly choose the initial weights
- (II) Select the first training pair
- (III) Forward computation that includes the following steps:
  - (A) Apply the inputs to the network
  - (B) Calculate the output for every neuron from the input layer, through the hidden layer (s), to the output layer
  - (C) Calculate the error at the outputs
- (IV) Backward computation
  - (A) Use the output error to compute error signals for preoutput layers
  - (B) Use the error signals to compute weight adjustments
  - (C) Apply the weight adjustments
- (V) Repeat Forward and Backward computations for other training pairs.
- (VI) Periodically evaluate the network performance. Repeat Forward and Backward computations until the network converges on the target output.

## 8 The structure of the developed system

Fig. 6 shows the structure of the developed intellectual breed recognition and evaluation of DA cattle.



**Fig. 6.** The structure of the developed intellectual system for breed recognition and cattle LW evaluation

## 9 Experiments and results

The problem of recognizing various breeds of cows and assessing their linear sizes with the subsequent determination of mass using a neural network predictive model based on a multilayer perceptron was solved. To recognize the breed of a cow, the Mask-RCNN network trained on the COCO sample was used, the source code of which is freely available on the Internet [25]. The network weights were fine-tuned on 250 images of representatives of each of the considered breeds of cows (Ayrshire, Holstein, Jersey, Red Steppe). To increase the size of the training and test samples, the augmentation method was used [26], which made it possible to obtain additional images from the original ones. The network was retrained over 5000 epoches using the SGD algorithm. To train the network, we used the Nvidia GeForce 1060 graphics card, which allowed us to speed up the learning process many times as compared to learning on the CPU. The results of recognition and evaluation of the size and weight of cows are given in Table 1. Examples of correct recognition are presented in fig. 7. It should be noted that, despite the rather extensive training sample and the achieved high recognition accuracy (92%), as a result of the network, false recognition occurs. An example of such recognition is shown in Fig. 8.

The obtained masks of recognized objects were used to determine their linear dimensions using the triangulation method (epipolar geometry). As additional information that allows us to adjust the obtained sizes, we used images from a 3D camera, which allow us to estimate the distance to the recognized object. Examples of such images are presented in Fig. 9.

**Table 1.** The obtained results

Breed	Ayrshire	Holstein	Jersey	Red Steppe
Average calculated dimensions (length of the body / chest girth) (cm)	149/171	155/180	132/160	157/189
Qualification weight (kg)	610	745	422	490
The standard deviation of obtained sizes	1.3	1.6	2.4	1.8
The mean absolute error (cm)	0.9	1.2	2.5	1.5
The standard deviation of absolute error (cm)	0.5	1.7	4.5	3.8
The average relative error	0.1	0.12	0.17	0.15
The standard deviation of the relative error	0.11	0.15	0.09	0.09
Number of correct recognition	461	418	424	439
Number of false positives	39	82	76	61
Accuracy	0.92	0.84	0.85	0.88
The ratio of false positives to the correct ones	0.09	0.2	0.18	0.14

**Fig. 7.** Examples of correct recognitions

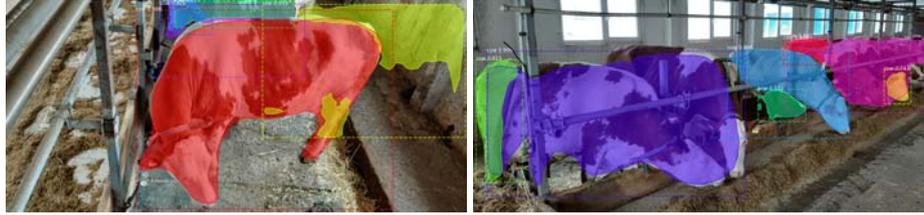


Fig. 8. Examples of false recognitions

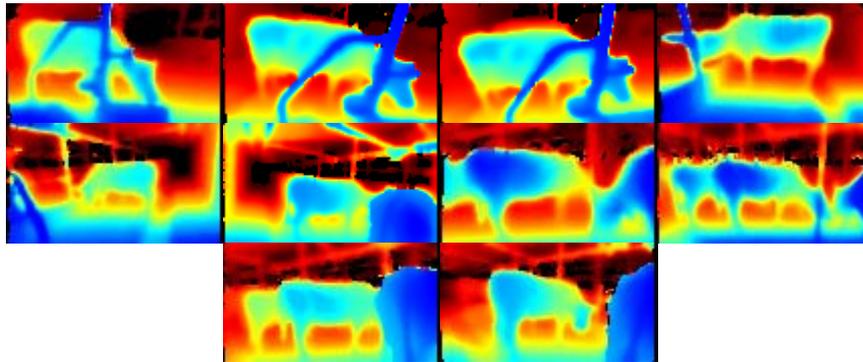


Fig. 9. Examples of 3D images from the camera

## Acknowledgement

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



## 10 Conclusions

The proposed method for measuring cattle using neural network image processing algorithms allows modern farmers to quickly and accurately measure the weight of the animal, as well as recognize its breed, saving time and reducing effort without compromising the habitat and disturbing livestock growth. This is possible because the method allows studying cows both in corrals and pastures, without interfering with their normal behavior. The proposed approach can be used to replace traditional methods that use direct observation and measurement, which adversely affect the behavior of animals. The system can be widely used in the management of modern farming. The accuracy and performance of the proposed methodology were tested

with the participation of the experts. The same measurements were carried out by the farm staff who confirmed the effectiveness of the proposed system.

## References

1. Tasdemir, S.: Determination of Body Measurements On the Holstein Cows by Digital Image Analysis Method and Estimation of Their Live Weight. Ph. D. thesis, Selcuk University, Konya, Turkey (2010).
2. Adamczyk, K., Molenda, K., Szarek, J., & Skrzyński, G.: Prediction of Bulls'slaughter Value From Growth Data Using Artificial Neural Network. *Journal of Central European Agriculture*, 6 (2), pp. 133-142 (2005).
3. Akilli, A., Atil, H.: Artificial Intelligence Technologies in Dairy Science: Fuzzy Logic and Artificial Neural Network. *Hayvansal Uretim*, 55 (1), pp. 39-45 (2014).
4. Akkol, S., Akilli, A., Cema, I.: Comparison of Artificial Neural Network and Multiple Linear Regression for Prediction of Live Weight in Hair Goats. *YYU Yuzuncu Yil Universitesi Journal of Agricultural Sciences*, 27 (1), pp. 21-29 (2017).
5. Ali, M., Eydurani, E., Tariq, M.M., Tirink, C., Abbas, F., Bajwa, M.A., ... & Shah, S.H.: Comparison of artificial neural network and decision tree algorithms used for predicting live weight at post weaning period from some biometrical characteristics in Harnai sheep. *Pakistan J. Zool.*, Vol. 47 (6), pp. 1579-1585 (2015).
6. Mortensen, A.K., Lisowski, P., Ahrendt, P.: Weight prediction of broiler chickens using 3D computer vision. *Computers and Electronics in Agriculture*, 123, pp. 319-326 (2016).
7. Raja, T.V., Ruhil, A.P., Gandhi, R.S.: Comparison of connectionist and multiple regression approaches for prediction of body weight of goats. *Neural Computing and Applications*, 21(1), pp. 119-124 (2012).
8. Salawu, E.O., Abdulaheem, M., Shoyombo, A., Adepeju, A., Davies, S., Akinsola, O., Nwagu, B.: Using artificial neural network to predict body weights of rabbits. *Open Journal of Animal Sciences*, 4 (04), 182 p. (2014).
9. Szyndler-Nędza, M., Eckert, R., Blicharski, T., Tyra, M., Prokowski, A.: Prediction of carcass meat percentage in young pigs using linear regression models and artificial neural networks. *Annals of Animal Science*, 16 (1), pp. 275-286 (2016).
10. Wang, Y., Yang, W., Winter, P., Walker, L.: Walk-through weighing of pigs using machine vision and an artificial neural network. *Biosystems Engineering*, 100 (1), pp. 117-125 (2008).
11. McNitt, J.I.: *Livestock Husbandry Techniques*, Low priced edition. Granada publishing company limited, 280 p. (1983).
12. Hartley, R.I., Zisserman, A.: *Multiple View Geometry*. Cambridge University Press (2004)
13. Bradski, G., Kaehler, A.: *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, 580 p. (2008).
14. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1 (4), pp. 541-551 (1989).
15. Fukushima, K.: Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biological Cybernetics* 36(4) (1980).
16. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. *arXiv: 1703.06870* (2017).
17. LeCun, Y., Bengio Y.: Convolutional networks for images, speech, and timeseries. *The Handbook of Brain Theory and Neural Networks*, pp. 255-258 (1995).

18. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In ICML, pp. 807-814 (2010).
19. Bodyanskiy, Ye.V., Rudenko, O.G.: Iskusstvennyye neyronnyye seti: arkhitektura, obucheniye, primeneniye. Khar'kov: TELETEKH, 372 p. (2004).
20. Uossermen, F.: Neyrokomp'yuternaya tekhnika. M.: Mir, 184 p. (1992).
21. Ham, F.M., Kostanic, I.: Principles of Neurocomputing for Science and Engineering, NY: Mc Graw-Hill Inc., 468p. (2001).
22. Patterson, D. Artificial Neural Networks, Theory and Application. Singapur: Prentice Hall Inc., 497p. (1996).
23. Rudenko, O.G., Bessonov, A.A.: Mnogokriterial'naya optimizatsiya evolyutsioniruyushchikh setey pryamogo rasprostraneniya. Problemy upravleniya i informatiki, №6, pp.29-41 (2014).
24. Cilimkovic, M.: Neural Networks and Back Propagation Algorithm, Institute of Technology Blanchardstown, Dublin, Ireland (2015).
25. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. [https://github.com/matterport/Mask\\_RCNN](https://github.com/matterport/Mask_RCNN)
26. Image augmentation for machine learning experiments. <https://github.com/aleju/imgaug>