

A method of measuring cattle parameters using neural network methods of image processing was proposed. To this end, several neural network models were used: a convolutional artificial neural network and a multilayer perceptron. The first is used to recognize a cow in a photograph and identify its breed followed by determining its body dimensions using the stereopsis method. The perceptron was used to estimate the cow's weight based on its breed and size information. Mask RCNN (Mask Regions with CNNs) convolutional network was chosen as an artificial neural network.

To clarify information on the physical parameters of animals, a 3D camera (Intel RealSense D435i) was used. Images of cows taken from different angles were used to determine the parameters of their bodies using the photogrammetric method.

The cow body dimensions were determined by analyzing animal images taken with synchronized cameras from different angles. First, a cow was identified in the photograph and its breed was determined using the Mask RCNN convolutional neural network. Next, the animal parameters were determined using the stereopsis method. The resulting breed and size data were fed to a predictive model to determine the estimated weight of the animal.

When modeling, Ayrshire, Holstein, Jersey, Krasnaya Stepnaya breeds were considered as cow breeds to be recognized. The use of a pre-trained network with its subsequent training applying the SGD algorithm and Nvidia GeForce 2080 video card has made it possible to significantly speed up the learning process compared to training in a CPU.

The results obtained confirm the effectiveness of the proposed method in solving practical problems

Keywords: image processing, convolutional network, multilayer perceptron, stereopsis, predictive model

UDC 004.852

DOI: 10.15587/1729-4061.2021.247648

BREED RECOGNITION AND ESTIMATION OF LIVE WEIGHT OF CATTLE BASED ON METHODS OF MACHINE LEARNING AND COMPUTER VISION

Oleksandr Bezsonov

Doctor of Technical Sciences, Professor*

Oleh Lebediev

PhD, Associate Professor**

Valentyn Lebediev

Postgraduate Student**

Yuriy Megel

Doctor of Technical Sciences, Professor, Head of Department

Department of Cybernetics

Kharkiv Petro Vasilenko National Technical University of Agriculture

Alchevskykh str., 44, Kharkiv, Ukraine, 61002

Dmytro Prochukhan

Lecturer

Kharkiv Computer Technology Professional College of the National

Technical University «Kharkiv Polytechnic Institute»

Manizera str., 12, Kharkiv, Ukraine, 61002

Oleg Rudenko

Corresponding author

Doctor of Technical Sciences, Professor, Head of Department*

E-mail: oleg.rudenko@nure.ua

*Department of Computer Intelligent Technologies and Systems***

Department of Electronic Computers*

***Kharkiv National University of Radio Electronics

Nauky ave., 14, Kharkiv, Ukraine, 61166

Received date 24.09.2021

Accepted date 22.11.2021

Published date 29.12.2021

How to Cite: Bezsonov, O., Lebediev, O., Lebediev, V., Megel, Y., Prochukhan, D., Rudenko, O. (2021). Breed recognition and estimation of live weight of cattle based on methods of machine learning and computer vision. *Eastern-European Journal of Enterprise Technologies*, 6 (2 (114)), 64–74. doi: <https://doi.org/10.15587/1729-4061.2021.247648>

1. Introduction

Image analysis using computer technology and various predictive applications are commonly used in various fields of human activity and agriculture is one of them. The number of farms is constantly growing and their productivity is increasing, so the importance of computer technology in the automation of agricultural processes is gradually rising. When raising cows, the relationship between live weight (LW), milk yield, and fodder consumption can be taken as criteria for organizing the attendance and nutrition of animals in present-day conditions. These parameters are quite important and must be strictly controlled. When they go beyond the permissible limits, it immune system

of cows is significantly affected, and, accordingly, the economic efficiency of the farm suffers. Negative changes in live weight can indicate health problems of the animal, inappropriate environmental conditions, and nutrition overlook. Consequently, such a parameter as live weight is certainly important for dairy cows [1]. It should be noted that the measurement and maintenance of cattle are still manual and rather expensive.

Therefore, it seems appropriate to use methods of computational intelligence to solve these problems, in particular, a neural network approach using several neural network models. Convolutional artificial neural networks are used to recognize cows in a photograph and identify their breed followed by determining its body dimensions using the ste-

reopsis method. Multilayer perceptron, in turn, is used to estimate cattle weight based on breed and size information.

To clarify information on the physical parameters of animals, a 3D camera (Intel RealSense D435i) is additionally used. It should be noted that the use of just 3D cameras does not provide good results due to their limited capabilities. Thus, images of cows taken from different angles were used to determine parameters of cow bodies using the photogrammetric method. Parameters such as withers height (WH), hip height (HH), body length (BL), and hip width (HW) of cows are obtained using photogrammetry. Using these parameters (input parameters WH, HH, BL, HW, and output parameter LW), an estimation model based on ANN was constructed.

The cow body size is determined by analyzing images of animals taken with synchronized cameras from different angles. First, a cow is identified in the image and its breed is determined using the Mask RCNN convolutional neural network. Then withers height, hip height, body length, and hip width of the cow are determined using the stereopsis method which makes it possible to determine geometric parameters of objects in digital images and take measurements on them. Acquisition of digital images and their photogrammetric processing involve several fully defined steps that enable the generation of 3D or two-dimensional digital models of the animal body. The resulting breed and size data are then fed to a predictive model that determines the animal weight.

Live weight of different animals is usually [1, 2] found for various purposes from regression equations derived with a considerable time input.

To do this, any method of determining the animal LW is used.

These methods, first of all, include the Trukhanovsky method in which the following formula is used to determine the live weight of adult cattle:

$$LW = \frac{A \cdot B}{100} \cdot K, \tag{1}$$

where *A* is the chest girth measured behind the shoulder blades (Fig. 1);

B is the straight body length measured with a stick, cm;

K is the correction factor (2 for dairy breeds and 2.5 for dairy-and-meat and meat breeds).

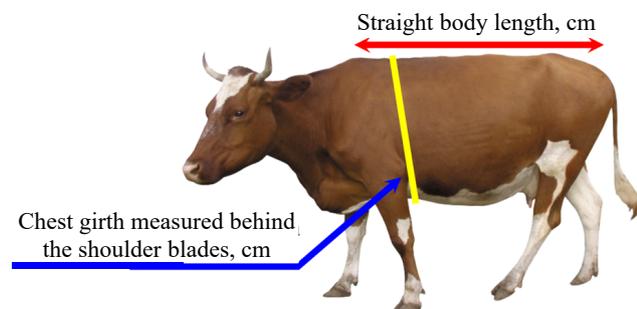


Fig. 1. Trukhanovsky method of measuring cattle

To determine the approximate live weight, special tables are used. Their initial data include animal measurements taken at a correct positioning of the animal (legs are upright and head at the back level).

There are other methods of estimating the cow weight and body size, in particular, the Pin Bone method [3]

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

where *BW* is the estimated live weight of livestock (pounds); *HG* is the withers height (inches); *BL* is body length (inches).

The conventional process of measuring an animal is shown in Fig. 2.



Fig. 2. Conventional process of measuring animals

Thus, when determining (evaluating) the LW of cattle, an important problem of determining its size arises. As shown in Fig. 2, this task becomes harder when measuring animals in real conditions.

Therefore, it seems very important to develop a modern approach to solving these problems based on computational intelligence, in particular, machine learning. This reduces both the time spent makes possible automation of these processes.

2. Literature review and problem statement

Practical approaches to predicting the live body weight of animals use various methods of regression analysis (simple regression, multiple regression, ridge regression, multivariate adaptive spline regression). The method of principal components is also used. Smart analysis methods are more flexible than classical ones [2].

Computational intelligence technologies have begun to be widely used in problems of improving the efficiency of agricultural production. For example, the problem of predicting the cost of bull slaughtering based on data of bull height using ANN was solved in [4]. Comparison of the effectiveness of neural networks and regression models of predicting the animal weight was made and a conclusion on a significant advantage of the former was drawn in [5, 6]. The problem of estimating the animal weight using methods of digital image analysis was considered in [7] which was a significant step forward.

The effectiveness of predicting the breeding value of dairy cattle using artificial neural networks and neuro-fuzzy systems was demonstrated in [8].

The effectiveness of artificial neural networks and decision trees used to predict live weight was compared and the effectiveness of neural networks was confirmed in [9]. The use of machine vision methods for solving such problems turned out to be very successful. For example, the weight of broiler poultry [10] and goats [11] was predicted using 3D computer vision. An artificial neural network has been successfully used in [12] for predicting the rabbit body weight. The percentage of carcass meat in young pigs was predicted

in [13] using linear regression models and artificial neural networks. Weighing pigs using machine vision and artificial neural networks was considered in [14] and a 3D shape of live pigs was obtained in [15] using stereophotogrammetry. The study [16] reported digital image analysis to estimate the live weight of pigs and mature live weight was estimated using body measurements in kara sheep [17].

Classification and forecast of milk yield of Holstein-Friesian cattle were considered in [18] using parametric and nonparametric statistical classification models. Comparison of the most common discriminant analysis strategy with a neural network approach showed that ANN is more effective in terms of predicting the level of milk yield.

However, all these studies [1–18] have a serious drawback: the animal size was measured either in real conditions using conventional analysis methods or digital images. This is explained by the fact that conventional methods using direct observation of animals and measurement of their parameters often negatively affect the behavior of animals. In addition, the time spent for obtaining a forecast increases significantly since all animals are measured sequentially and analysis of digital images requires high-quality photographs.

An approach in which ANN was used to estimate the weight of animals depending on the photogrammetric size of their body has turned out to be more effective [19]. Photogrammetry is a technology for obtaining quantitative and qualitative information on the natural and industrial characteristics of an object under study using photographic or non-photographic images [20]. Finding digital images and their photogrammetric processing enable the creation of 3D or 2D digital body models as an end product. The main disadvantage of this method consists in the need for high-quality images.

Measuring the points in images is much easier than in real life. Thus, it is possible to obtain many points in a very short time and build the best 3D models of objects [21]. Comparison of photogrammetric and classical methods of volume calculation showed that the photogrammetric method is more effective in terms of accuracy and cost of obtaining results. In addition, the application of this method significantly saves the study time [22]. However, as already noted, the approach used in [21, 22] requires high-quality images.

It should be specially pointed out that only one breed was considered and, as already noted, conventional methods of measuring their parameters were used in almost all these works [1–19]. The task becomes much more complicated if animals of various breeds differing in their parameters and weight enter the herd under consideration.

3. The aim and objectives of the study

The study objective is to develop a method of measuring the cattle parameters using neural network algorithms of image processing and structure of the intelligent system for recognizing the breed and estimating the cattle's live weight. This will make it possible not only to speed up the process of measuring the animal parameters and improve results due to the elimination of subjectivity but also to automate the management of the farm in general.

To achieve this objective, the following tasks were set:

- develop an algorithm of determining the cattle size based on the stereopsis method;
- develop a neural network system for animal detection and recognition of its breed;
- develop a structure of an intelligent system of breed recognition and estimation of livestock LW;
- carry out modeling of the process of recognizing various cow breeds and estimating their linear dimensions followed by determining their weight applying a neural network predictive model.

4. The study materials and methods

The following assumptions were made in the studies:

1. Different cow breeds are recognized (Ayrshire, Holstein, Jersey, Krasnaya Stepnaya) simultaneously.
2. To determine the cattle size, the stereopsis method is used consisting of the underwritten.

Let there are two cameras specified by their matrices P and P' in a certain coordinate system. In this case, it is said that there are a couple of calibrated cameras. If the centers of the cameras do not coincide, then this pair of cameras can be used to determine the 3D coordinates of the observed points.

The coordinate system is often chosen so that the camera matrices are of the following form: $P=K[I|0]$, $P'=K'[R'|t']$ (this can always be done by choosing the origin coinciding with the center of the first camera and directing the Z-axis along its optical axis).

Let us consider a point P in real 3D space projected simultaneously at two image points p and p' through two camera projection centers (C and C'). Points P , p , p' , C , and C' lie in a plane called the epipolar plane. The epipolar plane intersects each image to form the intersection lines. These lines (l and l') correspond to the projection of rays through p and P as well as p' and P and are called epipolar lines. This projection is described by epipolar geometry [23] (Fig. 3).

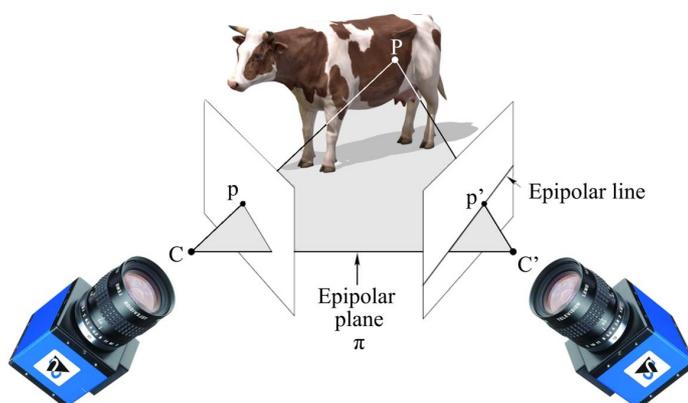


Fig. 3. Epipolar geometry

Epipolar geometry is used both to search for stereo pairs and check that a pair of points can form a stereo pair (i. e., a projection of some space point).

Epipolar geometry has a very simple coordinate notation. Let there be a pair of calibrated cameras and let p be homogeneous coordinates of a point in the image of one camera

and p' be that of the second camera. There is a 3×3 matrix F such that a pair of p, p' points is a stereo pair if and only if:

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

Matrix F is called a fundamental matrix of rank 2. It is determined with accuracy up to a nonzero factor and depends only on matrices of initial cameras P and P' .

In the case when forms of the camera matrices are $P=K[I|0]$, $P'=K'[R|t]$, the fundamental matrix can be calculated from the formula:

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

where notation $[e]_x$ for the vector e is calculated as follows:

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

Various algorithms for calculating F from a set of points are considered in [24]. In particular, algorithms of gradient descent of the Newton and Levenberg-Marquardt method are considered.

Equations of epipolar lines are calculated using the fundamental matrix. For point x , the vector defining the epipolar line will have form $l = Fp$ and equation of the epipolar line itself: $l^T p' = 0$. Similarly for point p' , the vector defining the epipolar line, will have the form: $l = F^T p'$.

The resulting epipolar lines are used to build depth maps. A depth map is an image in which distance to the camera is stored for each pixel instead of color. For this purpose, for each point in one image, a point paired with it is sought in another image. Besides, coordinates of their prototype in 3D space can be triangulated and determined for a pair of corresponding points. Knowing the 3D coordinates of the preimage, depth can be calculated as the distance to the camera plane [23].

3. To implement the recognition system, a convolutional neural network (CNN) is used.

Convolutional neural network (CNN) [25, 26] uses pattern recognition technology. It is based not on rigid algorithms laid down by developers but on training the system based on sequential identification of a huge number of images.

Subsequently, R-CNN networks (Regions with CNNs) were built on the basis of CNN to apply CNNs to the task of detecting all objects of mentioned classes and defining a constraint frame for each of them (object detection). R-CNN creates constraint frames for each object in the image or suggestions of regions using a selective search process. Fast R-CNN which increased the R-CNN performance classifies objects of each region along with tighter constraint frames. The next network, Faster R-CNN has improved the mechanism of generating candidate regions used in it by calculating regions not from the original image but from a feature map obtained from the CNN. For this purpose, a module has been added called Region Proposal Network (RPN).

Finally, the Mask R-CNN network develops the Faster R-CNN architecture [27, 28] by adding one more branch that predicts the position of the mask covering the found object and thus solves the instance segmentation problem. A detailed description of the network is given in [29] and a method of expanding the functionality of the Faster R-CNN network through the use of an additional image segmentation module is shown in Fig. 4. When receiving an image, the network issues objects (bbox) that constraint frames, classes, and masks. Mask R-CNN replaces the RoI pool layer with RoIAlign in the Faster R-CNN network structure. This method removes the coarse quantization of RoIPool and precisely aligns the retrieved objects with the input data.

It should be noted that Mask R-CNN is the fastest network now. In addition, the network with the Mask R-CNN architecture makes it possible to highlight contours ("masks") of instances of different objects in photographs. Even if there are several such instances, they have different sizes and are partially overlapped. This was the reason for choosing this network to solve the posed problem.

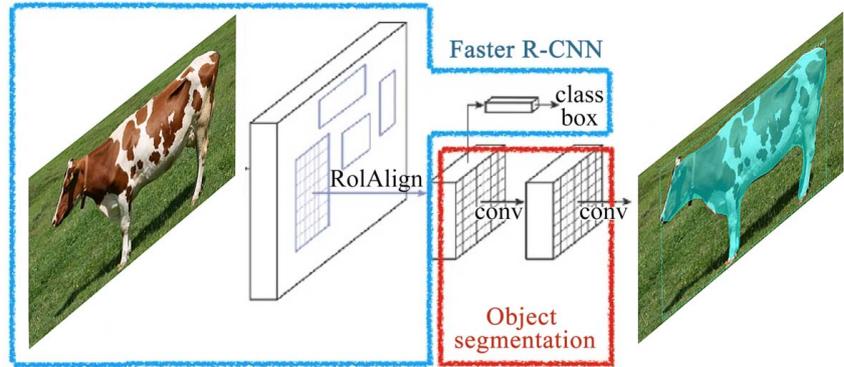


Fig. 4. Expansion of the Faster R-CNN network functionality

The Mask-RCNN network has the following advantages [28, 29]:

- it is one of the best ANNs for image recognition and classification;
- it has a much smaller number of adjusted weights compared to a fully connected ANN;
- parallelization of computations and, consequently, implementation of the algorithms of the network work and its training in graphics processors are possible;
- relative resistance to rotation and shift of the image being recognized;
- it is trained using the classic error backpropagation algorithm.

The architecture of the convolutional neural network

The structure of the convolutional neural network was initially designed taking into account structural features of some parts of the human brain responsible for vision. The development of such networks is based on three mechanisms:

- local perception;
- formation of layers in a form of a set of feature maps (shared weights);
- sub discretization (subsampling).

In accordance with these mechanisms, three main layers are used to build a convolutional neural network: convolution, pooling (subsampling or sub discretization), and a fully connected layer.

Convolution layer. The convolution formula for the l -th ($l=1, \dots, L$) layer of the network has the form [25]

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

and reflects the kernel motion ω^l along the input image for this layer of the image or the feature map y^{l-1} where $y_{ij}^{l-1} = f(x_{ij}^{l-1})$ is the image after the $(l-1)$ -th layer; $f(\bullet)$ is the used activation function; b^l is offset; i, j, a, b are indices of elements in matrices; s is the value of the convolution step.

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

Convolution preserves spatial relationships between pixels.

Each convolutional layer is followed by a sub discretization (subsampling) layer or a computational layer which serves to reduce the image dimension by locally averaging the neuron outputs.

Sub-sampling layer (pooling, MAX-pooling). The subsampling layer reduces the scale of planes by locally averaging the neuron outputs. Thus, a hierarchical organization is achieved. Subsequent layers extract more general characteristics less dependent on image distortion.

The pooling layer is described by the expression

$$x_{ij}^l = \sum_{a=-\infty}^{\infty} \sum_{b=-\infty}^{\infty} \beta^l \text{down}(\omega_{ab}^l \cdot y_{(is-a)(js-b)}^{l-1} + b^l), \tag{7}$$

where $\text{down}(\bullet)$ is the pooling function. This function sums up blocks of the input image, thus decreasing the dimensionality of the output layer image. Moreover, any output map is specified by two displacement parameters: multiplicative β^l and additive b^l .

The difference between the subsampling and convolution layers consists in that the regions of neighboring neurons overlap in the convolution layer which does not take place in the subsampling layer.

The pooling layer works independently of the cut-off of the input data depth and scales the volume spatially using a maximum function.

It is generally accepted in the convolutional network architecture to consider that presence of a feature is more important than the information on its location. Therefore, a maximum neuron is selected from several neighboring neurons in the feature map and its value is considered one neuron in the map of features of lower dimensionality.

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

A non-linear layer of activation. In these layers within the network, a non-linear 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 size.

Usually, due to significant positive properties, ReLu ($\text{ReLU}(x)=\max(0; x)$) and its various modifications (Leaky ReLU, Parametric ReLU, Randomized ReLU) are used as the activation function $f(\bullet)$ for hidden layers [30]. In a fully connected layer, the SoftMax function is used (when solving the classification problems):

$$f_j^L = e^{x_j^L} \left(\sum_{i=1}^{N^L} e^{x_i^L} \right)^{-1}$$

or a linear function (for regression problems). SoftMax function was used in our problem.

Dropout layer. Various regularization methods are used to avoid network retraining.

Dropout is a simple and effective method of regularization consisting in that a subnet is randomly and repeatedly singled out from its aggregate topology in the process of the network training. Thus, some of the neurons are turned off from the process and the next update of weights occurs only within the singled-out subnet. Thus, the weights change only for the remaining neurons. Each neuron is excluded from the aggregate network with a certain probability called the dropout coefficient.

This layer reduces the time of one training epoch due to the smaller number of parameters to be optimized and also makes it possible to better deal with the network retraining in comparison with standard regularization methods.

Normalizing layer. Standard normalization of inputs occurs in this layer (the sample mean of their values is subtracted and the result is divided by the root of the sample variance). The sampling values are calculated taking into account the values at this layer inputs in previous training iterations. This approach makes it possible to increase the speed of network training and improve the final result.

Fully connected layer. This layer is a conventional multilayer perceptron. Its purpose consists of classification. It simulates a complex nonlinear function. Its optimization improves the recognition quality.

Neurons of each map of the previous subsampling layer are connected 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 subsampling layer.

Like in conventional neural networks, neurons in a fully connected layer are connected with all activations in the previous layer. Their activations can be calculated by multiplying matrices and applying an offset.

The difference between fully connected and convolutional layers consists in that neurons of the convolution layer:

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

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

In practice, a quadratic function, cross-entropy, or some combined functional is used as a criterion of the error function.

Multilayer perceptron. A backpropagation neural network (BPNN) is a multi-layer direct communication neural network that uses a supervised learning algorithm known as an error backpropagation algorithm. Errors accumulated in the output layer are propagated back to the network to correct the weights.

Training of SNN is similar to the training of direct propagation ANN and consists in correcting its weight parameters based on minimizing a certain selected loss function: quadratic function or cross-entropy is commonly used.

Both standard error backpropagation method and its various modifications can be used to train convolutional neural networks. Derivation of the backpropagation algorithm for CNN training is considered in detail in [31]. This method is based on the algorithm of stochastic gradient descent (SGD).

As usual, the most common neural network learning algorithm based on the gradient descent method (error backpropagation method) and its modifications including SGD, Adam, AdaGrad, AdaDelta, etc. are used in practice [32–34].

These algorithms are also used to train the multilayer perceptron.

4. The Mask R-CNN neural network [34] is implemented based on TensorFlow, a popular open platform for machine learning. The network was significantly accelerated through the use of an NVIDIA GeForce RTX 2080 graphics card. Its use requires the installation of an additional library of GPU-accelerated primitives called cuDNN.

5. To increase the size of training and test samples, the augmentation method was used [36].

6. To clarify information on the physical parameters of an animal, two synchronized Intel RealSense D435i 3D cameras are used. The included drivers and accompanying software should be installed for their effective work.

5. The results obtained in studying the effectiveness of the neural network-based method of cattle breed recognition and live weight estimation

5.1. Development of an algorithm for determining the cattle size

An algorithm for determining the cattle size based on the stereopsis method and epipolar geometry has been developed. It includes the following steps:

1. Detection of a cow using a convolutional ANN.
2. 3D coordinates of observed points of the object are determined using two calibrated cameras.
3. Fundamental matrix is calculated using formula (4).
4. Equations of epipolar lines are calculated.
5. Depth maps are built which are then used to determine the animal size.

5.2. Development of a neural network system of animal detection and recognition of its breed

The developed neural network system of animal detection and recognition of its breed uses Mask Regions with CNNs (MRCNN) network pre-trained on the Common Objects in Context (COCO) sample. Its source code is freely available [34].

This network is the fastest network now. In addition, the network with the Mask R-CNN architecture makes it possible to highlight contours (“masks”) of instances of different objects in photographs. Even if there are several such instances, they have different sizes and partially overlap. This was the reason for choosing this network to solve the set problem.

Detailed Mask R-CNN architecture is shown in Fig. 5. It should be noted that the ResNet-101 network was used as a CNN (backbone) in Mask R-CNN (101 means the number of layers).

To obtain a matrix (“map”) of features of the entire image, a small (3×3 and 5×5 pixels) kernel sequentially shifted horizontally and vertically is used.

A feature map of 25×25 size can be obtained from the original image having the size of 128×128 pixels using the RoIAlign procedure.

To calculate the feature values, bilinear interpolation for four nearest integer points is used.

The procedure of detecting objects using the R-CNN network includes the following steps:

- highlighting the candidate regions with the help of Selective Search;
- converting the region to a size acceptable by the CNN;
- obtaining a 4096-dimensional feature vector using the CNN;
- carrying out N binary classifications of each feature vector using N linear SVMs;
- a linear regression of the region frame parameters for more accurate coverage of the object.

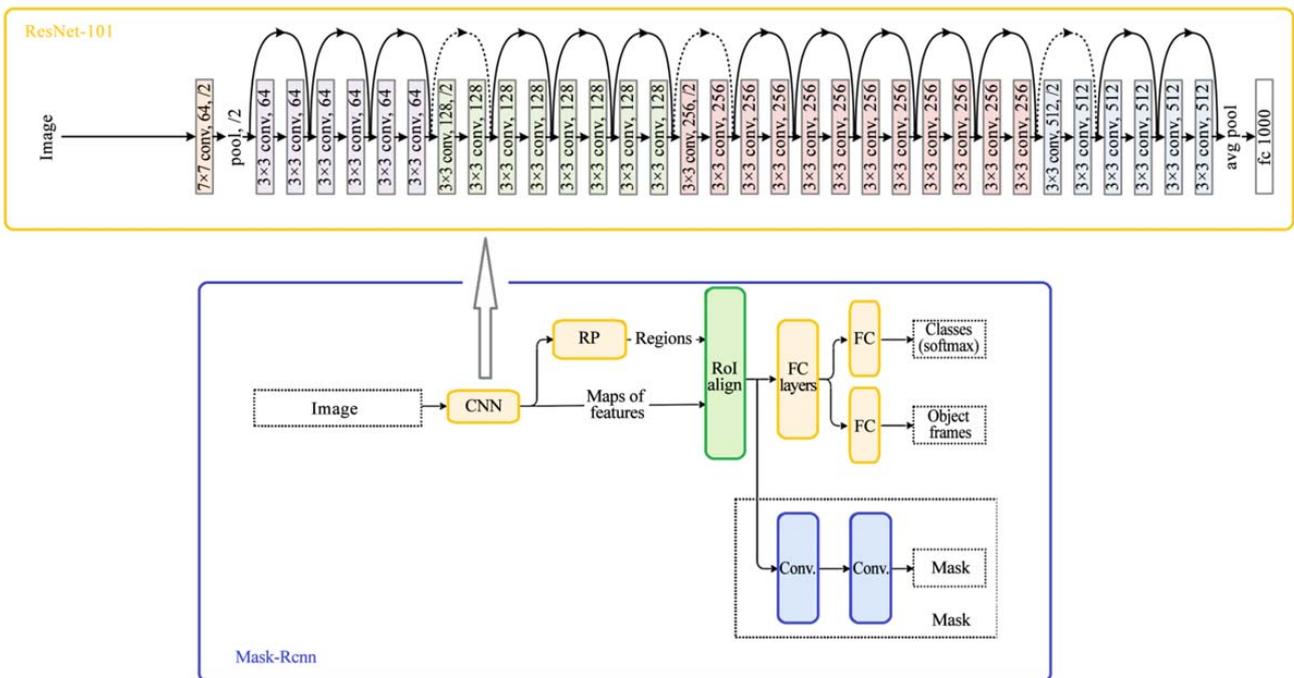


Fig. 5. Detailed architecture of the Mask R-CNN network

The process was accelerated in the Fast R-CNN network due to passing through the CNN not each of the 2000 candidate regions individually but the whole image. The proposed regions are then superimposed on the resulting general feature map. The features found in different regions were converted to a fixed size, i.e., the feature matrix for a candidate region was calculated using the RoIPooling procedure. The region window having width w and height h was broken into a grid with HW cells of $h/H w/W$ size. Max Pooling was applied to each of such cells to select only one value giving the resulting HW feature matrix.

5. 3. Developing the structure of an intelligent system of cattle breed recognition and live weight estimation

The structure of the designed intelligent system is shown in Fig. 6.

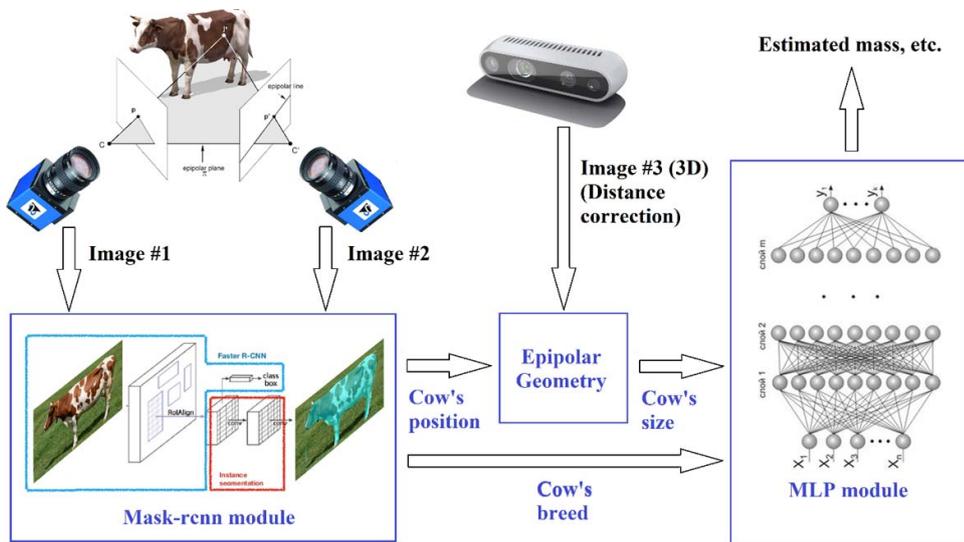


Fig. 6. Structure of the intelligent system of cattle breed recognition and live weight estimation

As seen from the figure, this system contains synchronized 3D cameras and neural network modules (Mask R-CNN and multilayer perceptron (MLP)).

The animal image taken simultaneously by two cameras located at some distance from each other is fed to the first module (Mask R-CNN) which serves to identify the breed and determine the animal position (coordinates). The position of the animal is refined using information from the 3D camera in the epipolar geometry module which calculates the animal size. Information about the animal size and breed is transmitted to the second module (MLP) to assess the animal weight and other parameters necessary for correct diet formulation.

5. 4. Modeling the process of recognizing different breeds of cows and estimating their linear dimensions

The process of recognition of various cow breeds and estimation of their linear dimensions was modeled with subsequent mass determination using a neural network predictive model.

Examples of correct and erroneous cow recognition using a pre-trained network are shown in Fig. 7, 8. In this case, test images taken in ideal conditions were used. It should be noted that the pre-trained network cannot recognize the cow breed. It can just separate the class of cows from other classes of objects. In this regard, in order to use the pre-trained network in a real system capable of recognizing the breed of a cow

and estimating its live weight, the network was further trained using 250 images. Representatives of each cow breed under consideration (Ayrshire, Holstein, Jersey, Krasnaya Stepnaya) were used for this aim. To increase the size of the training and test samples, the augmentation method was used [35] which made it possible to obtain additional images from original ones. The network was additionally trained in 5,000 epochs using the SGD algorithm [32]. Nvidia GeForce 2080 video card was used to train the network which has made it possible to speed up the learning process many times compared to training in a CPU. The learning errors are shown in Fig. 9: classification error (a), localization error (b), general error (c), regularization error (d). The results of recognition and estimation of the size and weight of cows are shown in Table 1. An example of correct recognition of cows in real living conditions is shown in Fig. 10. It should be noted that, despite a fairly large training sample and the achieved high

recognition accuracy (92 %), false recognition occurs as a result of network operation. An example of such recognition is shown in Fig. 11.

The resulting masks of recognized objects were used to determine their linear dimensions using the triangulation method (epipolar geometry). We used images taken with a 3D camera as additional information allowing us to correct the obtained dimensions which has made it possible to estimate the distance to the recognized object. An example of such an image is shown in Fig. 12.



a



b

Fig. 7. Examples of correct recognition of test images: a – one animal; b – several animals



Fig. 8. Examples of false recognition of test images: *a* – one animal was recognized simultaneously as a cow and as a giraffe; *b* – one animal was recognized simultaneously as a cow and as a horse

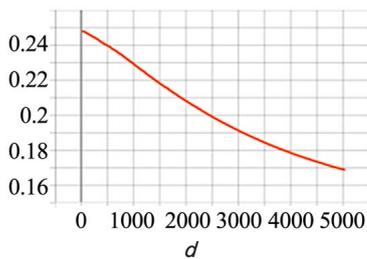
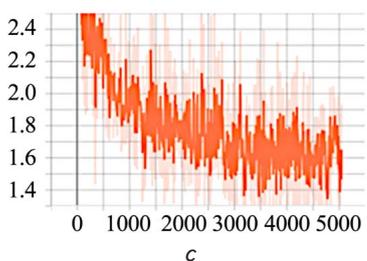
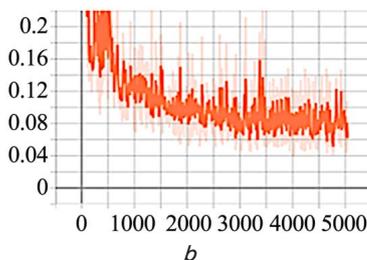
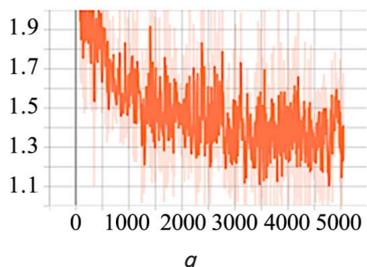


Fig. 9. Learning errors: *a* – classification error; *b* – localization error; *c* – general error; *d* – regularization error



Fig. 10. An example of correct recognition in real conditions



Fig. 11. Examples of false recognitions in real conditions: *a* – false detection; *b* – several animals were recognized as one

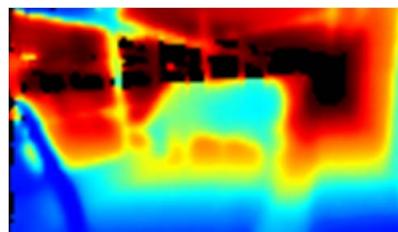


Fig. 12. An example of an image taken with a 3D camera

Table 1
Results obtained using the proposed method

Breed	Ayrshire	Holstein	Jersey	Krasnaya Stepnaya
Average calculated measurements (oblique torso/chest) (cm)	149/171	155/180	132/160	157/189
Weight estimate (kg)	610	745	422	490
Standard deviation of obtained dimensions	1.3	1.6	2.4	1.8
Mean absolute error (cm)	0.9	1.2	2.5	1.5
Standard deviation of absolute error (cm)	0.5	1.7	4.5	3.8
Average relative error	0.1	0.12	0.17	0.15
Standard deviation of relative error	0.11	0.15	0.09	0.09
Number of correct recognitions	461	418	424	439
Number of false recognitions	39	82	76	61
Accuracy	0.92	0.84	0.85	0.88
Ratio of false to correct recognitions	0.09	0.2	0.18	0.14

As can be seen from Table 1, the percentage of correct recognitions of cow breeds in real conditions is quite high and amounts to 84–92 %. However, despite the sufficiently large training sample and the high recognition accuracy achieved, false recognitions also arise as a result of network operation.

6. Discussion of the results obtained in studying the neural network method of recognizing cow breeds and their size estimation

In the proposed method of image recognition, the Mask-RCNN convolutional neural network was chosen as an ANN. This network was trained using the error backpropagation algorithm [34].

The developed algorithm of determining the cattle size using the stereopsis method makes it possible to study cows both in pens and on pastures without interfering with their usual behavior. The algorithm is implemented on the basis of information obtained using the Mask R-CNN convolutional ANN described by relations (6), (7) and shown in Fig. 4. This network solves the problem of breed recognition based on the analysis of images taken with two calibrated cameras. The animal size is determined from the obtained depth maps without direct human participation.

Ayrshire, Holstein, Jersey, and Krasnaya Stepnaya breeds were considered as cow breeds for recognition.

The use of a pre-trained network with its subsequent additional training using the SGD algorithm and the use of an Nvidia GeForce 2080 video card has made it possible to speed up the learning process many times compared to learning in a CPU. To increase the size of training and test samples, the augmentation method was used [35]. All this has provided a significant reduction in time for obtaining a forecast since the animals were measured in parallel. An additional reduction in forecast time was provided by the possibility of parallelizing computations in the Mask-RCNN network. The change in the network learning errors shown in Fig. 9 indicates the effectiveness of the SGD algorithm.

The resulting masks of recognized ANN objects were used to determine their linear dimensions using the triangulation method (epipolar geometry). The use of additional information in the form of images from a 3D camera (Fig. 11) has allowed us to correct the obtained animal size.

The neural network system developed for detecting an animal and recognizing its breed shown in Fig. 5 used the Mask R-CNN pre-trained on a COCO sample. The ResNet-101 network was used as a CNN (backbone) in Mask R-CNN.

The structure of the developed intelligent system shown in Fig. 6 includes synchronized 3D cameras and neural network modules (Mask R-CNN and MLP) which provide animal recognition and prediction of weight. The first module serves to identify the animal breed and determine its position (coordinates). The animal position is refined using information from the 3D camera in the epipolar geometry module which calculates the animal size. Information on the animal size and breed is transmitted to the second (MLP) module to assess the animal weight and other parameters necessary to formulate the correct diet. Thus, the system provides automation of the process of recognizing the animal breed and determining its weight, eliminating inaccuracies and subjectivity inherent in existing approaches.

The modeling results presented in Fig. 7–12 and Table 1 indicate the effectiveness of the proposed approach expressed in high speed and accuracy of recognition. Thus, this method provides higher efficiency and quality of recognition and measurement of animal parameters in comparison with existing methods.

Advantage of the developed method in comparison with those known on this topic: it makes it possible to quickly

estimate the live weight of cattle in a herd consisting of animals of different breeds. Effective breed recognition is ensured by the use of a pre-trained Mask-RCNN ANN, one of the best image recognition and classification ANNs. Its further training using the SGD algorithm and Nvidia GeForce 2080 video card has made it possible to speed up the learning process many times compared to training in CPU [1–18].

Limitations of the proposed method: its implementation requires modern computational and technical means. To solve the problem and train the network, we used high-quality images of animals taken with a 3D Intel RealSense D435i camera. Shooting was carried out in good lighting conditions. Therefore, a high percentage (84–92 %) of correct cow breed recognition was obtained in real conditions. Recognition accuracy may vary for images of a different quality which requires additional studies.

In the future, work will continue to fill the database of reference animals. Also, the following activities are planned:

- assess the sensitivity of the developed animal recognition method under different shooting conditions and using different equipment;
- consider and clarify various options of using the proposed intelligent system for cattle breed recognition and live weight estimation to automate the information processing;
- train the developed convolutional network for other conditions of obtaining animal images;
- improve the structure of the convolutional network and procedures of its training;
- apply the approach to recognize other animal species.

7. Conclusions

1. An algorithm for determining the cattle size using the stereopsis method has been developed which has made it possible to study cows both in pens and on pastures without interfering with their usual behavior. Implementation of the algorithm is based on information obtained using a convolutional ANN which, in turn, uses images taken with two calibrated cameras. The animal size is determined from obtained depth maps with no direct human participation.

2. A neural network system for animal detection and breed recognition has been developed using a pre-trained Mask-RCNN convolutional neural network. To improve the efficiency of this network, it was additionally trained using the SGD algorithm and Nvidia GeForce 2080 video card. This has made it possible to speed up the learning process many times compared to training in CPU as used in [1–18].

3. Structure of an intelligent system of breed recognition and live weight estimation including synchronized 3D cameras and neural network modules has been developed. Modules (Mask R-CNN and MLP) provide breed recognition and animal weight prediction. Mask R-CNN identifies the animal breed and determines its position (coordinates). The position of the animal is refined using information from the 3D camera in the epipolar geometry module used to calculate the animal dimensions. Weight and other parameters necessary for drawing up a correct animal diet are estimated in the MLP module based on information on the animal size and breed. Thus, the system provides automation of the process of animal breed recognition and determination of live weight eliminating inaccuracies and subjectivity inherent in existing approaches.

4. Simulation modeling of the process of recognizing various breeds of cows and estimation of their linear dimensions with subsequent weight determination using a neural network predictive model has been carried out. Ayrshire,

Holstein, Jersey, and Krasnaya Stepnaya breeds were considered as cow breeds for recognition. The percentage of correct recognition of cow breeds in real conditions is quite high and amounts to 84–92 %.

References

1. Tasdemir, S., Urkmez, A., Inal, S. (2011). Determination of body measurements on the Holstein cows using digital image analysis and estimation of live weight with regression analysis. *Computers and Electronics in Agriculture*, 76 (2), 189–197. doi: <https://doi.org/10.1016/j.compag.2011.02.001>
2. Celik, S., Eyduran, E., Karadas, K., Tariq, M. M. (2017). Comparison of predictive performance of data mining algorithms in predicting body weight in Mengali rams of Pakistan. *Revista Brasileira de Zootecnia*, 46 (11), 863–872. doi: <https://doi.org/10.1590/s1806-92902017001100005>
3. McNitt, J. I. (1983). *Livestock Husbandry Techniques*. London: Granada publishing company limited, 288.
4. Adamczyk, K., Molenda, K., Szarek, J., Skrzyński, G. (2005). Prediction of Bulls'slaughter Value From Growth Data Using Artificial Neural Network. *Journal of Central European Agriculture*, 6 (2), 133–142. Available at: https://www.academia.edu/22135262/Prediction_of_BullsSlaughter_Value_From_Growth_Data_Using_Artificial_Neural_Network_Przewidywanie_Warto%C5%9Bci_Rze%C5%BAnej
5. Akkol, S., Akilli, A., Cemal, İ. (2017). Comparison of Artificial Neural Network and Multiple Linear Regression for Prediction of Live Weight in Hair Goats. *Yüzüncü Yıl Üniversitesi Tarım Bilimleri Dergisi*, 27 (1), 21–29. doi: <https://doi.org/10.29133/yyutbd.263968>
6. Khorshidi-Jalali, M., Mohammadabadi, M. R., Esmailzadeh, A., Barazandeh, A., Babenko, O. I. (2019). Comparison of Artificial Neural Network and Regression Models for Prediction of Body Weight in Raini Cashmere Goat. *Iranian Journal of Applied Animal Science*, 9 (3), 453–461. Available at: http://ijas.iaurasht.ac.ir/article_667543_108967f61ea69baf983318670f81ffd.pdf
7. Nicolas, F. F. C., Saludes, R. B., Relativo, P. L. P., Saludes, T. A. (2018). Estimating live weight of philippine dairy buffaloes (*bubalus bubalis*) using digital image analysis. *Philipp J Vet Anim Sci*, 44 (2), 129–138. Available at: <https://www.pjvas.org/index.php/pjvas/article/view/207/183>
8. Shahinfar, S., Mehrabani-Yeganeh, H., Lucas, C., Kalhor, A., Kazemian, M., Weigel, K. A. (2012). Prediction of Breeding Values for Dairy Cattle Using Artificial Neural Networks and Neuro-Fuzzy Systems. *Computational and Mathematical Methods in Medicine*, 2012, 1–9. doi: <https://doi.org/10.1155/2012/127130>
9. Ali, M., Eyduran, E., Tariq, M. M., Tirink, C., Abbas, F., Bajwa, M. A. et. al. (2015). 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.*, 47 (6), 1579–1585. Available at: [http://zsp.com.pk/pdf47/1579-1585%20\(10\)%20QPJZ-0146-2015%2014-7-15%20REVISEDVERSION_FINAL.pdf](http://zsp.com.pk/pdf47/1579-1585%20(10)%20QPJZ-0146-2015%2014-7-15%20REVISEDVERSION_FINAL.pdf)
10. Mortensen, A. K., Lisouski, P., Ahrendt, P. (2016). Weight prediction of broiler chickens using 3D computer vision. *Computers and Electronics in Agriculture*, 123, 319–326. doi: <https://doi.org/10.1016/j.compag.2016.03.011>
11. Raja, T. V., Ruhil, A. P., Gandhi, R. S. (2011). Comparison of connectionist and multiple regression approaches for prediction of body weight of goats. *Neural Computing and Applications*, 21 (1), 119–124. doi: <https://doi.org/10.1007/s00521-011-0637-z>
12. Salawu, E. O., Abdulraheem, M., Shoyombo, A., Adepeju, A., Davies, S., Akinsola, O., Nwagu, B. (2014). Using Artificial Neural Network to Predict Body Weights of Rabbits. *Open Journal of Animal Sciences*, 04 (04), 182–186. doi: <https://doi.org/10.4236/ojas.2014.44023>
13. Szyndler-Nędzka, M., Eckert, R., Blicharski, T., Tyra, M., Prokowski, A. (2016). Prediction of Carcass Meat Percentage in Young Pigs Using Linear Regression Models and Artificial Neural Networks. *Annals of Animal Science*, 16 (1), 275–286. doi: <https://doi.org/10.1515/aoas-2015-0057>
14. Wang, Y., Yang, W., Winter, P., Walker, L. (2008). Walk-through weighing of pigs using machine vision and an artificial neural network. *Biosystems Engineering*, 100 (1), 117–125. doi: <https://doi.org/10.1016/j.biosystemseng.2007.08.008>
15. Wu, J., Tillett, R., McFarlane, N., Ju, X., Siebert, J. P., Schofield, P. (2004). Extracting the three-dimensional shape of live pigs using stereo photogrammetry. *Computers and Electronics in Agriculture*, 44 (3), 203–222. doi: <https://doi.org/10.1016/j.compag.2004.05.003>
16. Wongsriworaphon, A., Arnonkijpanich, B., Pathumnakul, S. (2015). An approach based on digital image analysis to estimate the live weights of pigs in farm environments. *Computers and Electronics in Agriculture*, 115, 26–33. doi: <https://doi.org/10.1016/j.compag.2015.05.004>
17. Yilmaz, H. M., Yakar, M., Yildiz, F. (2008). Digital Photogrammetry in Obtaining of 3D Model Data of Irregular Small Objects. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Beijing, 125–130. Available at: https://www.isprs.org/proceedings/XXXVII/congress/3b_pdf/23.pdf
18. Radwan, H., Qaliouby, H., Elfadl, E. (2020). Classification and prediction of milk yield level for Holstein Friesian cattle using parametric and non-parametric statistical classification models. *Journal of Advanced Veterinary and Animal Research*, 7 (3), 429. doi: <https://doi.org/10.5455/javar.2020.g438>

19. Tasdemir, S., Ozkan, I. A. (2019). Ann approach for estimation of cow weight depending on photogrammetric body dimensions. *International Journal of Engineering and Geosciences*. doi: <https://doi.org/10.26833/ijeg.427531>
20. Cooper, M. A. R., Robson, S. (1996). Theory of Close Range Photogrammetry. *Close Range Photogrammetry and Machine Vision*, 9–51.
21. Yilmaz, H. M. (2010). Close range photogrammetry in volume computing. *Experimental Techniques*, 34 (1), 48–54. doi: <https://doi.org/10.1111/j.1747-1567.2009.00476.x>
22. Yakar, M., Yilmaz, H. (2008). Using in volume computing of digital close range photogrammetry. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Beijing, 119–124. Available at: https://www.isprs.org/proceedings/xxxvii/congress/3b_pdf/22.pdf
23. Hartley, R., Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press. doi: <https://doi.org/10.1017/cbo9780511811685>
24. Bradski, G., Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, 580.
25. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1 (4), 541–551. doi: <https://doi.org/10.1162/neco.1989.1.4.541>
26. LeCun, Y., Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, 255–258.
27. Girshick, R. (2015). Fast R-CNN. 2015 IEEE International Conference on Computer Vision (ICCV). doi: <https://doi.org/10.1109/iccv.2015.169>
28. Wang, J., Ye, Z. (2018). An improved faster R-CNN approach for robust hand detection and classification in sign language. *Tenth International Conference on Digital Image Processing (ICDIP 2018)*. doi: <https://doi.org/10.1117/12.2503080>
29. He, K., Gkioxari, G., Dollar, P., Girshick, R. (2017). Mask R-CNN. 2017 IEEE International Conference on Computer Vision (ICCV). doi: <https://doi.org/10.1109/iccv.2017.322>
30. Nair, V., Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. *Proceedings of the 27 th International Conference on Machine Learning*.
31. Zhang, Z. (2016). Derivation of Backpropagation in Convolutional Neural Network (CNN). Available at: [https://zzutk.github.io/docs/reports/2016.10%20-%20Derivation%20of%20Backpropagation%20in%20Convolutional%20Neural%20Network%20\(CNN\).pdf](https://zzutk.github.io/docs/reports/2016.10%20-%20Derivation%20of%20Backpropagation%20in%20Convolutional%20Neural%20Network%20(CNN).pdf)
32. Raimi, B. K. (2015). 10 Gradient Descent Optimisation Algorithms + Cheat Sheet. Available at: <https://www.kdnuggets.com/2019/06/gradient-descent-algorithms-cheat-sheet.html>
33. Rudenko, O., Bezsonov, O., Oliinyk, K. (2020). First-Order Optimization (Training) Algorithms in Deep Learning. *Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2020)*. Volume I: Main Conference. Lviv, 921–935. Available at: <http://ceur-ws.org/Vol-2604/paper61.pdf>
34. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. Available at: https://github.com/matterport/Mask_RCNN
35. Image augmentation for machine learning experiments. Available at: <https://github.com/aleju/imgaug>