Available at www.sciencedirect.com

INFORMATION PROCESSING IN AGRICULTURE 1 (2014) 73–81

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

Real-time recognition of sows in video: A supervised approach

Ehsan Khoramshahi [a,*], *Juha Hietaoja* [a], *Anna Valros* [b], *Jinhyeon Yun* [b], *Matti Pastell* [a]

[a] *Department of Agricultural Sciences, University of Helsinki, Finland*
[b] *Faculty of Veterinary Medicine, University of Helsinki, Finland*

A R T I C L E I N F O

Article history:
Received 19 May 2014
Received in revised form
8 July 2014
Accepted 18 July 2014
Available online 10 August 2014

Keywords:
Precision farming
Supervised classification
Real-time image-processing
Neural network

A B S T R A C T

This paper proposes a supervised classification approach for the real-time pattern recognition of sows in an animal supervision system (asup). Our approach offers the possibility of the foreground subtraction in an asup's image processing module where there is lack of statistical information regarding the background. A set of 7 farrowing sessions of sows, during day and night, have been captured (approximately 7 days/sow), which is used for this study. The frames of these recordings have been grabbed with a time shift of 20 s. A collection of 215 frames of 7 different sows with the same lighting condition have been marked and used as the training set. Based on small neighborhoods around a point, a number of image local features are defined, and their separability and performance metrics are compared. For the classification task, a feed-forward neural network (NN) is studied and a realistic configuration in terms of an acceptable level of accuracy and computation time is chosen. The results show that the dense neighborhood feature ($d.3 \times 3$) is the smallest local set of features with an acceptable level of separability, while it has no negative effect on the complexity of NN. The results also confirm that a significant amount of the desired pattern is accurately detected, even in situations where a portion of the body of a sow is covered by the crate's elements. The performance of the proposed feature set coupled with our chosen configuration reached the rate of 8.5 fps. The true positive rate (TPR) of the classifier is 84.6%, while the false negative rate (FNR) is only about 3%. A comparison between linear logistic regression and NN shows the highly non-linear nature of our proposed set of features.

1. Introduction

Nowadays, digital information in different forms, including digital images, has relentlessly covered many aspects of our daily lives. This rapid advancement is a natural result of Moore's law progression, as well as the establishment of standards in digital content [12].

Image processing (IP) has recently become an integral part of a variety of domains such as health, multimedia, agriculture, robotic, telecommunication, entertainment and many

* Corresponding author. Address: P.O. Box 28, Koetilantie 5, Department of Agricultural Sciences, University of Helsinki, Helsinki 00014, Finland. Tel.: +358 40 4444135.
E-mail address: ehsan.khoramshahi@helsinki.fi (E. Khoramshahi).
Peer review under the responsibility of China Agricultural University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.inpa.2014.07.002

others [13]. Among them, modern agriculture has many challenges that could be appropriately addressed by IP techniques. A major challenge in this field is continuous monitoring of animals, which has the potential to improve animal welfare and a higher production level. The continuous supervision of animals by farm workers is expensive and expected to become even more costly in the near future [15]. On the other hand, digital videos include a considerable amount of data, while they are cheap sources of information, so efficient conversion of this data into information would serve many applications in precision livestock farming (PLF), including supervisory tasks.

In fact, a number of behavior analyzers (such as aggression detection [18] and sleeping detection [7]) are potentially eligible to be designed in a way to accept image-based features as extra inputs. IP techniques are indeed the core part to handle the complexities and produce the required information for decision support routines of a higher level.

A well-designed image/video analysis routine can be considered as an expert routine that extracts information about the current status and activity of animals (such as position and movements) in order to predict upcoming events. This means that the quality of an IP algorithm directly affects the quality of consecutive predictor routines.

The role of automatic image-based monitoring is especially emphasized for high-risk situations (e.g. farrowing), where the lack of human supervision could lead to animal injuries or even death. In such cases, early warning from an image-based system allows an efficient intervention to improve the production performance and animal welfare. Therefore, IP techniques should be integrated into such predictive routines to help farms by taking care of individuals or groups of animals automatically.

The very first step in the process of designing a video-based animal behavior classifier is finding the animal's patterns efficiently. This problem could generally be considered as a supervised machine learning (ML) task, because there is prior geometric and/or radiometric knowledge about the underlying pattern, but it can also be treated as a semi-supervised approach by combining on-line estimations.

Many IP methods during the last decade have been employed to locate animals in videos. For example, the combination of likelihood ratios and shading was applied by Hu and Xin [9] to segment pigs from the background. Chen et al. [3,4] implemented an averaging/thresholding routine inside an FPGA to detect animals. Haar of Oriented Gradients (HOG) was employed by Zhang et al. [19] to capture the shape and texture features of the animal's head. Viazzi et al. [16] used the frame difference to extract the pig image. Ahrendt et al. [1] developed an image-based real-time tracking algorithm for pigs. They employ support map segments to build a Gaussian model of the individual pigs. A combination of fuzzy-c means clustering, morphological operation and blob analysis has been studied for segmentation by Kaiyan et al. [11]. To classify aggression behavior among pigs, Viazzi et al. [17] defined a set of images features for Motion History Image (MHI). They recognized the positive cases by employing Linear Discriminant Analysis (LDA).

During recent years, a group of IP methods have been developed and successfully employed in many real-time foreground/background classification tasks (see e.g. [6,8,20,21]).

Background subtraction has been widely studied and used in real applications since the late 1990s. The first versions were mainly based on simple averaging. Those approaches estimated the background by an average over the time and then subtracted the current image from the average to derive an estimation of dynamic objects. Subsequently, statistical approaches have been introduced which have noticeably improved the classification quality. They were mainly based on a normality assumption about the distribution of gray levels. Single Gaussian and the mixture of two Gaussian (MOG) were the very first models. Single Gaussian has been proved to work well, especially for indoor situations with a small change in illumination, but it was unable to work for those situations where the color density of the background and foreground was close. MOG has since been developed, and is now able to approximate more complicated distributions. Many branches from the first MOG have been introduced to improve the defects of the original algorithm. For example Friedman and Russell [6] improved it by computing the effect of shadow by adding a third cluster. He used the expectation–maximization (EM) technique, such as the classic MOG approach, in order to fit his clustering model to the data. His approach has been proved to work well for the situations where there is enough information for different labels (shadow, pattern, and background).

Despite the considerable progress that has been made in recent background-subtraction techniques, they mainly works well in conditions where the target objects are uniformly visible to the camera, and active enough to capture the movement variance (e.g. movements of cars or people), whereas their performance can be significantly affected when the scene becomes static, or the target pattern is covered by some fixed elements (such as crate's bars). In order to overcome this problem, we here propose an algorithm for the special condition of an animal supervision (asup) system that is able in work in static conditions where the target patterns of an asup are covered by some elements and divided into small non-connected pieces.

In our paper we examine the above specific situation by: proposing a set of image-based features according to local neighborhoods, investigating their separability, comparing them in terms of accuracy and required computational time for classification with a feed-forward neural network (ffd-NN), and selecting an optimized configuration for ffd-NN in order to stay in a computationally acceptable timing frame.

The motivation for our research is to prepare algorithmic and structural software platforms for automatic predictions of a domestic animal's behaviors. To enable this research, a C++ image-processing framework has been developed with the aim of reducing the complexity of design by deploying a dynamic-programming scheme.

The first part of this article is a preliminary description of the feature sets. The supervised classification scheme is proposed next. Finally, the set of features are compared in detail, and the process of finding the optimized feature set is discussed.

The main objective of this work is to propose a supervised scheme for the real-time detection of sows in live video streams. The contributions of this paper include: (1) comparing a set of image-based features for the pattern recognition task, (2) investigating their separability and selecting the most promising feature set, (3) finding an optimum configuration for the NN classifier in terms of acceptable accuracy and computational complexity.

2. Materials and methods

To develop an image-based system for predicting the farrowing of sows and alerting the farmer, a data set of 7 days/sow of indoor surveillance for 7 sows for 24 h during the day and night was recorded. Each single recording consists of a video of a sow kept in farrowing crate of (220×75 cm). In each farrowing session, 8 surveillance cameras (Tracer TS 6030PSC IR), with a capture rate of 12 fps, were connected via cables to a central recording unit (Philips SAA7130HL) at the height of 2.5 m. All the cameras were equipped with IR emitters, to monitor in poor lighting conditions, and were configured to capture frames with a resolution of 352×288 pel^2 (MPEG-4 codec). The video dataset was preprocessed with a PC in order to extract frames with a 20-s time shift, and filtered with a Gaussian kernel. We filtered a total of 2,890,163 frames. Among them, we selected a set of 215 frames from 7 different random sows with similar lighting conditions and color content as a training database for the supervised training process. All of the selected frames were marked by a human operator by filling the regions of interests with one or more tag colors. The data set was divided into three parts: a training, test, and cross-validation set with 70%, 20%, and 10% contributions, respectively. The PC was a HP EliteBook 2560p using an Intel Corei7 2620 M CPU at 2.7 GHz and 8 Gb of RAM.

The classification process is performed through the following steps:

- In order to avoid the over fitting problem due to similarities between structures along a line, a randomized process selects the locations of the training points (the density of the positive and the negative cases are considered to be different), and for each point a coordinate is calculated. The padding between positive and negative cases is an adjustable parameter that plays a key role in the skew condition of the problem.
- A preprocessing routine whitens the input data (sub sec. 3.4).
- A trainer process randomly initializes the weights (or loads the previous weights, in the case that they exist), and then calls the optimizer. When the optimizer needs the gradients of the network, it calculates them with the back-propagation algorithm. The optimizer trains the network using a set of pair-tagged images, each of which consists of an image and a corresponding map. The desired regions are indicated in the maps by filling with one or more tag color.
- A ffd-NN classifier process uses the forward-propagation algorithm to classify a new data set.

We used a logistic-regression classifier with the same inputs to compare the classification accuracy and investigate the non-linearity of our feature set. PCA technique has been employed for preprocessing task. Receiver operating characteristic (ROC) curve, in this paper, has been calculated by sorting the predictive scores, and walking over different decision thresholds. The area under the ROC curve has been used as an indicator to quantify the efficiency of an underlying classifier.

In order to make the design and testing easier, a C++ computational core, as well as a graphical user interface (GUI) has been developed. An IP algorithm has been considered as a directed acyclic graph (DAG), where nodes represent a process, sockets act as inputs and outputs, and directed edges are responsible for the data connection that acts as the pipes which allow the data to flow.

3. Feature sets review

This part proposes a promising set of simple features, that can be employed to separate a sow from the background. Fig. 1 illustrates a small neighbor-based topological structure around a point. Each structure is labeled with a 3-letter code that represents the windowing size, a sparse/dense structure, and the proportion of local points. Each of these features exposes special characteristics. The computational complexity of their classification is proportional to the number of points they contain. The best feature is well separable, and produces the best classification result, while needing less computational time. There is a trade-off between the non-linearity of a feature and the required computational time, which is considered in section [5].

4. Classification methodology

Choosing a classification method strongly depends on a set of factors, including the characteristics of a feature set, the dimensionality of the data, and size of the training set (Demšar [5]). In an easy case, when a feature set is linearly separable, a lightweight linear classifier, such as logistic regression, is able to separate class members.

We employed the logistic regression classifier and also a ffd-NN to separate the positive cases from the negatives. The decision boundary between positive and negative cases has been estimated by using a ffd-NN as a compact representation of a non-linear decision boundary for the set of continuous random variables.

5. Data analysis

5.1. *Preprocessing*

Preprocessing is a mandatory step that robustifies the estimation. It can be done during a two-step procedure. In the first step, the data matrix is constructed, and normalized to a zero-mean and unit-variance matrix, as follows:

$$\hat{X}^{(i)} = X^{(i)} - \bar{X} \tag{1}$$

$$\tilde{X}_j^{(i)} = \hat{X}_j^{(i)} / \sigma_j \tag{2}$$

Data points in Eq. (1) are inserted in column order inside the data matrix (X). So each column represents a pixel. The cardinality of this space, which the points lie in, is equal to ($3 \times n$), where (n) is the number of points that a structure contains.

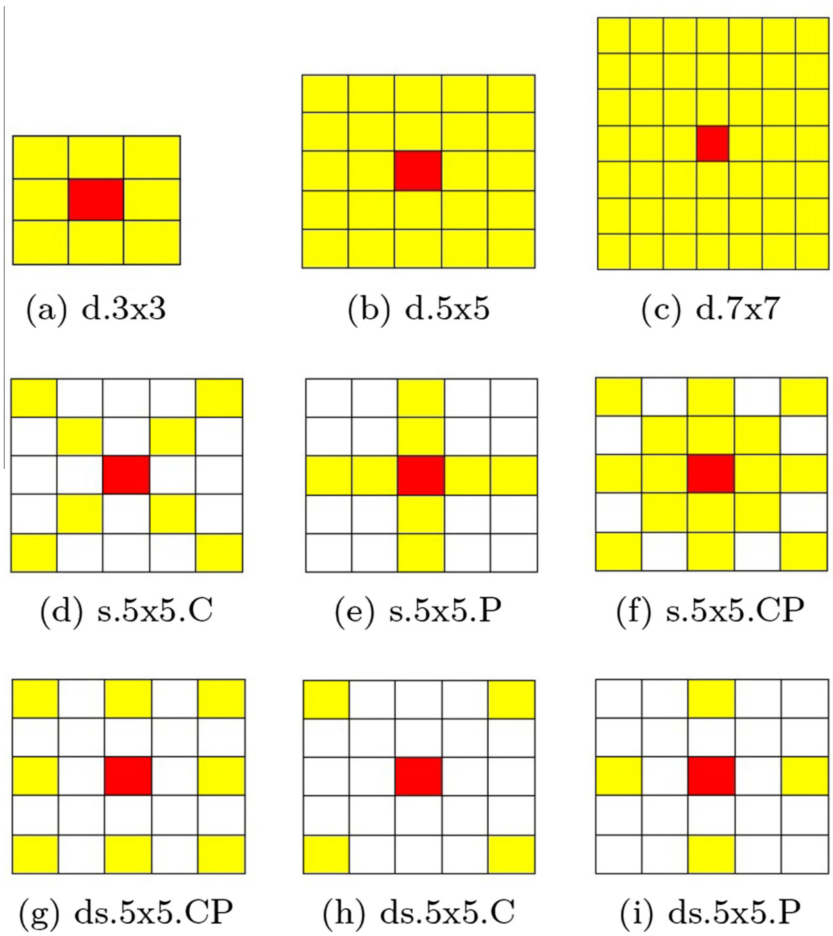

Fig. 1 – List of local features. The red square indicates the central point, and the yellow indicates cooperation of neighbor points in the structure. The items (a–c) are dense features (their labels started by 'd'), while the rest are sparse (their labels started by 's') and double sparse (their labels started by 'ds'). The features (e and i) are plus-shape (their labeled ended with '.P'), (d and h) are cross-shape (their labeled ended with '.C'), while features (f and g) contains both directions (their labeled ended with '.CP'). As we move from top to bottom, we can see that the number of features reduces.

In Eq. (1), X represents the mean value of the vector points, and $\tilde{X}_i^{(i)}$ is the jth component of the ith normalized data sample. The standard deviation along the jth axis is represented by σ_j.

In the second step, the resultant data will be whitened. Whitening is the process of making the data's distribution normal (Gaussian). Without this step, the classification problem would possibly suffer from an ill-posedness and singularity that instabilizes the classification accuracy. Whitening is addressed by principal component analysis (PCA). In order to perform PCA, the inner product (or covariance) matrix of the normalized data vectors (Eq. (2)) is calculated by:

$$C = \tilde{X}.\tilde{X}^T \tag{3}$$

Then the spectral decomposition of (C) is constructed as:

$$C = U.\Sigma.U^T \tag{4}$$

In the final stage, each point is projected on the orthonormal pc directions as:

$$X_W^{(i)} = \tilde{X}.^{(i)} \times U \tag{5}$$

In Eq. (4), the eigenvalues exist as the diagonal elements of (σ), and corresponding eigenvectors are stored as columns of (U).

Finally, the normalized data points of Eq. (5) are fed to the nodes of the ffd-NN input layer.

5.2. *Logistic regression*

In a classification with a logistic regression (LR) approach, the following hypothesis is assumed:

$$h_\theta(X) = g(\theta^T.X) \tag{6}$$

where $g(X)$ is the logistic function. Hypothesis of a LR (Eq. (6)), which is always in the interval (0,1), is assumed to represent the dependency of a sample point (X) to a class, where the decision boundary is linearly approximated. Basic LR is able to solve linear two-class classification problem, while some minor modification over the parameters makes it possible to estimate non-linear decision boundaries.

In order to classify the data with the LR classifier, the parameters (θ) of the hypothesis (h_θ (X)) are computed through minimizing the following cost function:

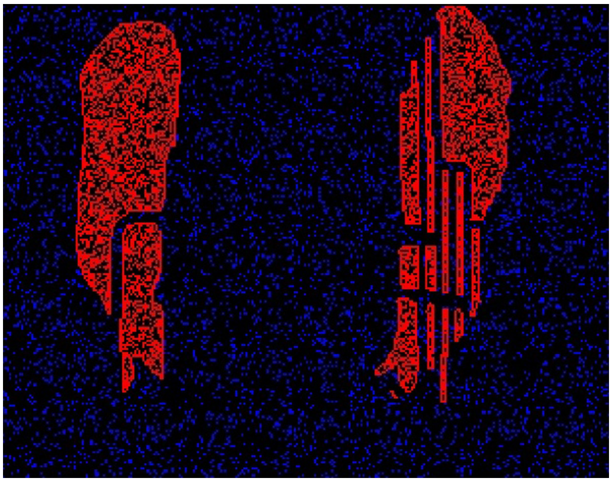

Fig. 2 – An example of the contributions of the positive and negative cases in the training set. Red points are the positive cases and blue points are the background.

$$J(\Theta) = -\frac{1}{m}\sum_{i=1}^{m}\mathrm{cost}(h_\theta(X), y^{(i)}) \quad (7)$$

$$\mathrm{Cost}(h_\theta(X), y^{(i)}) = \begin{cases} -\log(h_\theta(X)), & \text{if } y^{(i)} = 1 \\ -\log(1 - h_\theta(X)), & \text{if } y^{(i)} = 0 \end{cases} \quad (8)$$

Due to the nonlinear characteristic of the sigmoid function, the cost function for every individual sample of the logistic regression is modified as (Eq. (8)). Consequently, this modification makes the main cost function (Eq. (7)) convex. After approximating the optimum parameters, the decision is made based on the values of (h_θ (X)) such that the values less than a threshold (which is often 0.5) considered to belong to the first class, and similarly the higher values as the second class.

5.3. *Feed-forward neural network*

A feed-forward neural network (ffd-NN) is among the most powerful non-linear supervised classifiers. It is a branch of the perceptron algorithm, which can amazingly adapt to highly non-linear high-dimensional data. A well-optimized implementation of this classifier can be fast enough and acceptable for many real-time classification tasks, including background/foreground classification [10].

ffd-NN consists of layers of connected nodes. The first and last layers are called the input and output, respectively. The layers in between are named hidden. The topological structure of a ffd-NN allows the nodes of each layer to be connected to the nodes of only the next layer in the chain. Each layer contains a fixed additional element that is disconnected from the previous node and acts as a constant.

A ffd-NN could consist of several hidden layers. Adding more hidden layers theoretically increases the flexibility of the classifier by increasing the degrees of freedom, but in practice, it will make the classification process time consuming. This flexibility is handled by finding a trade-off between the computational complexity and the classification accuracy. The cost of the computation is an important factor in between, which logarithmically increases as a function of the parameters.

Inside a ffd-NN with one or more hidden layers, each node (in the hidden or output layer) applies the logistic function (Eq. (9)) to the summation of inputs, and sends the result (Eq. (10)) to the connected nodes (in next layer) [13], as follows:

$$g(x) = \frac{1}{1 + e^{-x}} \quad (9)$$

$$\alpha_j^l = g(\theta_{j0}^l.\alpha_0^{l-1} + \theta_{j1}^l.\alpha_1^{l-1} + \cdots + \theta_{jm}^l.\alpha_m^{l-1}) \quad (10)$$

Where α_j^l is the activation of node (*j*) in layer (l), and θ_{jk}^l is the corresponding weight of the connection between node (*j*) in layer (l) and node (*k*) in layer (l-1).

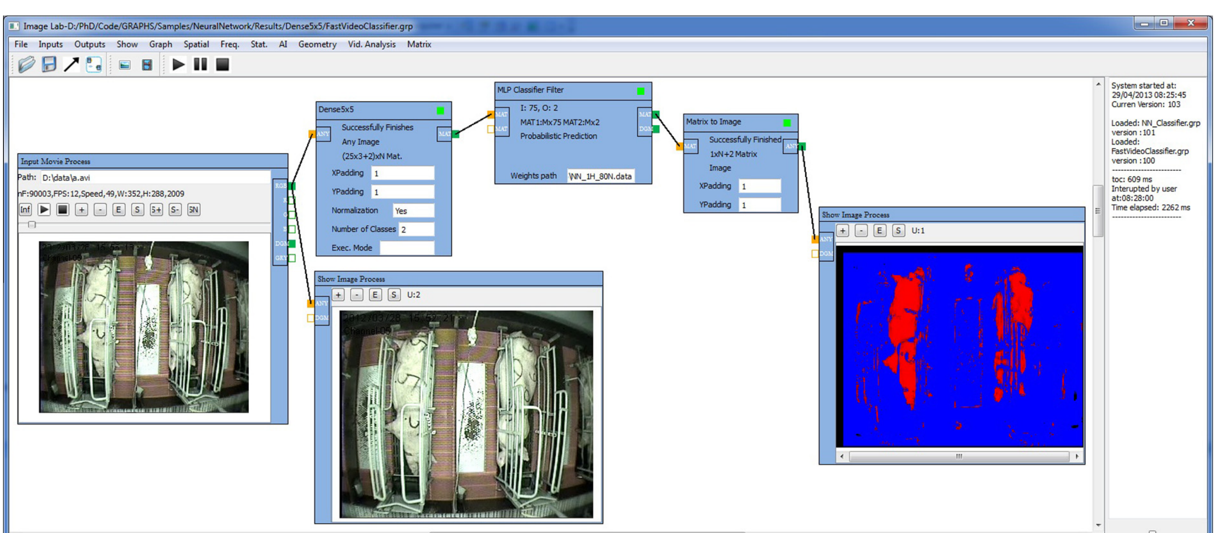

Fig. 3 – An example of the classification graph, which is executing inside the graph manager. The data flows from left to right, and are processed by every unit.

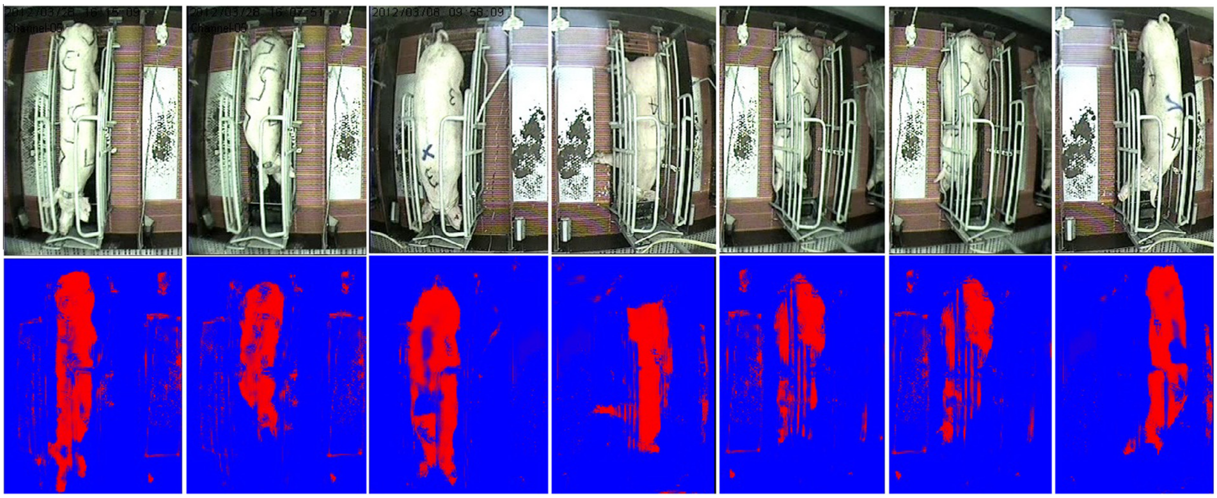

Fig. 4 – The examples of successful classifications cases. The first row contains images from a single sow. The second row contains the classification result. The sow patterns have been marked by the red color according to ffd-NN's output node value, whereas the background is marked by the blue.

The cost function for a ffd-NN is formulated according to Eq. (11):

$$J(\Theta) = -1/m\sum_{i=1}^{m}\sum_{k=1}^{K}y_k^i.\log h_\theta(x^{(i)})_k + (1-y_k^i).\log(1-h_\theta(x^{(i)})_k)) + reg(\Theta) \tag{11}$$

In the above equation, Θ represents the weights between the nodes (parameters), (m) represents the number of samples, (K) is the number of points, and ($reg\Theta$) stands for the regularization term regarding the parameters Θ.

5.4. *Training of the ffd-NN*

Despite the fact that the cost function of a ffd-NN is not convex [2], it is differentiable, and its gradients with respect to the parameters are efficiently calculated by the back-propagation algorithm. Most of the times, a local minimum of the cost of a ffd-NN regarding the parameters is an acceptable approximation for the parameters, so a gradient-based optimization technique is able to solve the training problem.

The training process is started by a random initialization of parameters, and then in each step, gradients of the cost function (Eq. (3)) are calculated. The trainer algorithm would then take a leap towards the location of the minimum by the following equation:

$$\Theta_j = \Theta_j - \alpha\frac{\partial}{\partial\Theta}J(\Theta) \tag{12}$$

Accelerated gradient descent [14] is a gradient-based optimization method that finds the location of a minimum (or a maximum) faster than the classic gradient descent approach

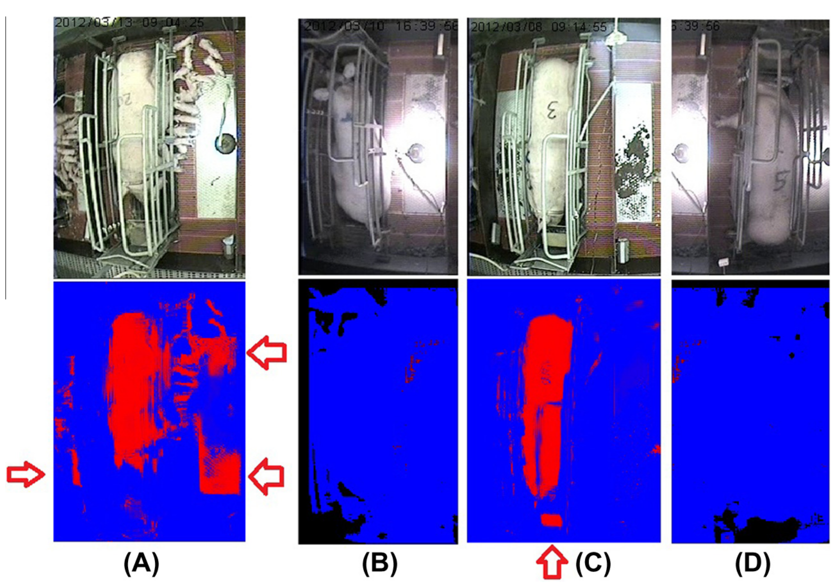

Fig. 5 – The examples of unsuccessful classifications cases. The first row contains images from a single sow. The second row contains the classification result. Outliers are marked by a red arrow.

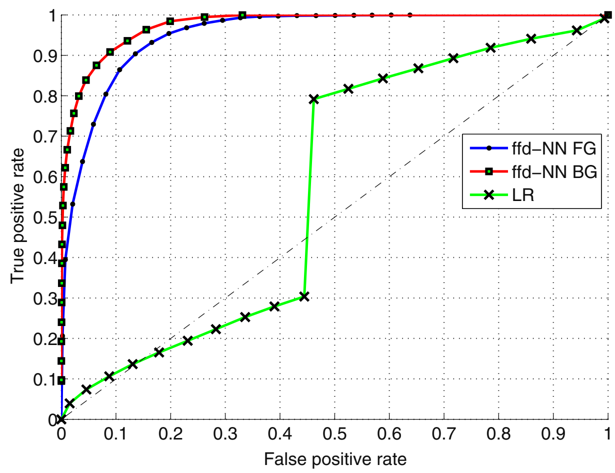

Fig. 6 – Comparison between logistic regression (LR) and feed-forward neural network (ffd-NN) for d.5 × 5 feature.

(Eq. (12)). Because of its convergence speed, it has been employed to train our ffd-NN classifier. The basic matrix optimization techniques, as well as the hardware-specific optimizations (SSE2 and SSE3), are employed to make the classification task faster.

5.5. *Classification by the ffd-NN*

After the training phase, the ffd-NN classifier is employed for the classification task (Fig. 3). The process of using the structure, topology, and weights of a ffd-NN in order to calculate the values of output nodes is called forward propagation. After this step, each output node will contain a number in the range [0,1] that shows the dependency of an input on each of the classes.

6. Results

Fig. 3 demonstrates the usefulness of a module-base data-processing framework for the algorithm-design phase. By employing this frame-work, the non-linearity of our proposed features was investigated by a comparison with a LR-based classifier. This comparison for the feature coding (d.3 × 3) is illustrated by Fig. 6. For each feature of Fig. 1, Table 1 lists the required number of calculations per point and the estimated required time for classification of the pixels of an image. A comparison between the proposed features has been also conducted and illustrated in Figs. 8 and 9. The set of neighbor-based local features was studied, and the progressive improvement in separability measures was observed (Figs. 10 and 11). The ROC of each of the feature codings for foreground and background nodes was separately calculated and compared (Figs. 8 and 9). AUC was also calculated as a performance measure, and compared. Fig. 4 demonstrates a few samples as representatives of high-quality classification cases, while Fig. 5 shows some of the low-quality cases (see Fig. 7).

7. Discussion

Most of the feature sets in Fig. 2 show non-linear properties for our data (Fig. 6), which implies the non-linearity of the optimal decision boundary. The main reason is that the desired pattern is very similar to components of our scene (such as lighting reflections, floor, bars). This problem could therefore be treated as a two-class classification problem with a non-linear decision boundary, where the size of the training set is approximately 100,000. In our experiment, a significant processing cost was introduced by adding more than one hidden layer, while the non-linearity of the classifier was not significantly improved. Thus, a three-layered ffd-NN has been chosen as the optimal structure, with an input layer, one hidden layer with 40 nodes, and an output layer. Because only two different classes exist (the sow and the background), it was possible to put only one node in the output layer and perform the inference according to (sub sec. 5.3). However, adding an additional node had considerable advantages. It robustified the estimation by introducing the additional degrees of freedom, and brought the possibility of estimating the background separately from the foreground, by introducing two different decision boundaries, while it did not produce a considerable computational overhead. By using

Table 1 – Performance comparison table. Topological neighbor-based structures are compared according to their computational complexity and classification performance.

No.	Future specific parameters				Statistical performance					
	Code	No. of elements	No. of flop per image[a]	Classification time per image[b] (ms)	Foreground TPR (%)	Foreground FPR (%)	Background TNR (%)	Foreground FNR (%)	Accuracy (%)	Precision F1 (%)
1	d.3 × 3	9 × 3	2320	191	92.1	4.1	95.9	7.9	93.9	94.1
2	d.5 × 5	25 × 3	6160	307	94.2	3.4	96.9	5.8	95.3	95.6
3	d.7 × 7	49 × 3	11,920	483	94.2	3.0	97.0	5.8	95.5	95.7
4	s.5 × 5.C	9 × 3	2320	191	74.4	3.0	97.0	25.6	92.9	79.0
5	s.5 × 5.P	9 × 3	2320	191	65.9	3.8	96.2	34.1	90.7	71.8
6	s.5 × 5.CP	17 × 3	4240	238	75.9	3.5	96.5	24.1	92.8	79.0
7	ds.5 × 5.C	5 × 3	1360	170	68.7	3.7	96.3	31.3	91.4	74.1
8	ds.5 × 5.C	5 × 3	1360	170	59.1	4.1	95.9	40.9	89.3	66.5
9	ds.5 × 5.CP	9 × 3	2320	191	64.0	4.1	95.9	36.0	90.2	70.1

a Floating-point operation.
b For an image of size 352 × 288.

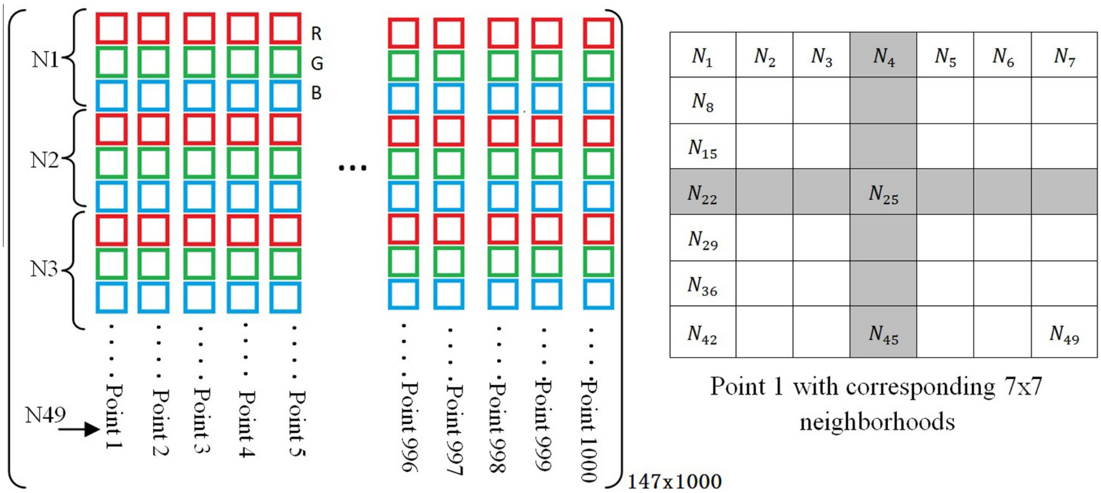

Fig. 7 – An example of a vectorized input with 147 features for the (d.7 × 7) structure.

this additional node, it is also possible to estimate 'unknown' cases that are dissimilar to both the foreground and the background training samples.

The classification precision obviously increased by moving from the sparse structures towards the dense ones. The dense structures (d.3 × 3, d.5 × 5, and d.7 × 7) show acceptable performance when they were coupled with the non-linear classifier. Generally, among the sparse features ('s' and 'ds' codings), those with a higher density of points showed better classification performance, while they had lower computational complexity. In equal situations, the dense structures showed a higher ability in the foreground detection than the sparse counterparts. Among the sparse structures, (C) and (CP) codings showed better performance than (P) coding. Generally, by increasing the number of local points, the precision of detecting the foreground (animal pattern) increases with a fairly considerable rate, while the improvement rate of background detection almost stays constant. This means that by increasing the feature density, the ability to detect the sow patterns will be positively increased, while the ability to detect the foreground will stay almost constant.

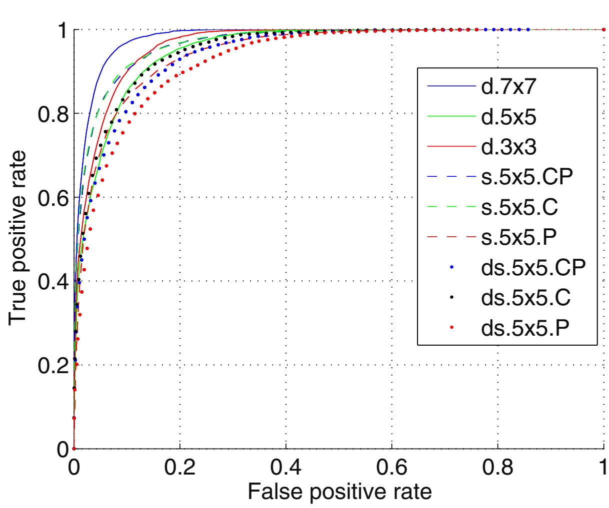

Fig. 8 – ROC of NN's foreground node, NN config.: {3*num. pixels, 40, 2}.

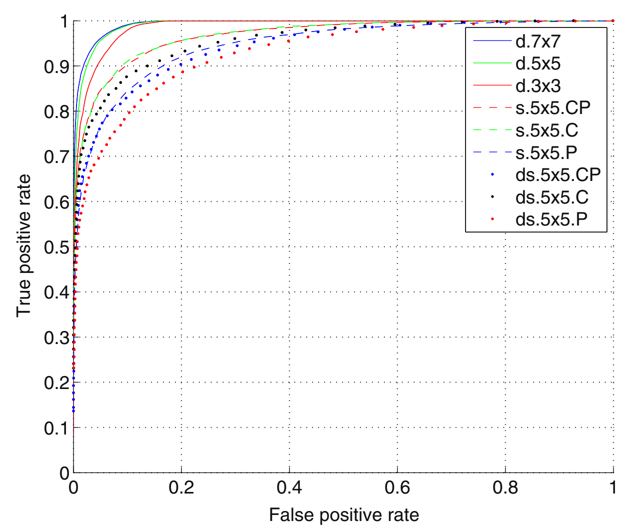

Fig. 9 – ROC of NN's background node, NN config.: {3*num. pixels,40,2}.

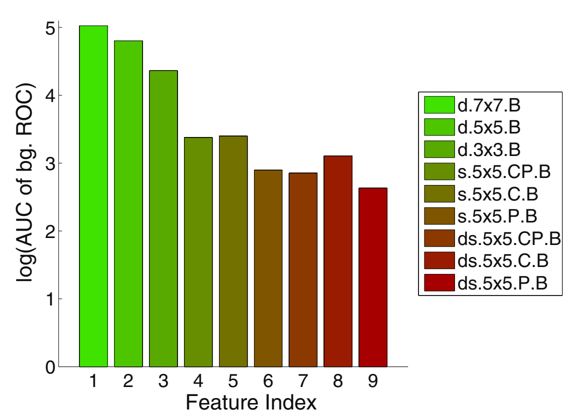

Fig. 10 – Log(AUC) of the background node for all the feature codings.

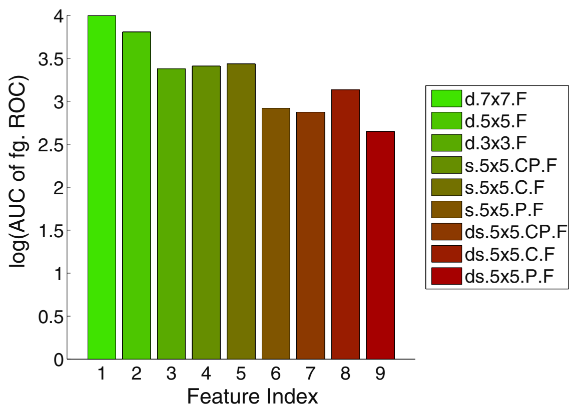

Fig. 11 – Log(AUC) of the foreground node for all the feature codings.

Despite the fact that (s.5 × 5.C), (s.5 × 5.P) and (ds.5 × 5.CP) have the same number of points as (d.3 × 3), they showed a lower performance for foreground detection, so the 3 × 3 dense structure (d.3 × 3) is selected as the most economical structure, which showed acceptable classification accuracy and precision, while it is suitable for real-time applications, and can be employed within an acceptable time frame (processing more than 4 frames per second).

After the training phase, the network efficiently extracts the pig image, even in complicated occluded scenes, where the whole body is not visible to the camera (Fig. 4).

8. Conclusion

Our supervised classification algorithm (the optimum feature coupled with the optimum ff-NN configuration) was proved to work with an acceptable performance in the static condition of an asup, where the other statistical approaches (such as MOG) significantly suffered from a lack of information on the distribution of the background, and were unable to perform the classification tasks. Despite the benefits of employing a ffd-NN in a foreground-background subtraction task, it is affected by the over fitting to the training condition, that caused the network to be unable to classify the black–white images that captured under IR illumination (part (B) and (D) in Fig. 5). For further studies, we suggest to employ a conditional classification scheme, instead of a single classifier, to enable the system to select a correct classifier according to a set of probabilistic criteria.

REFERENCES

[1] Ahrendt P, Gregersen T, Karstoft H. Development of a real-time computer vision system for tracking loose-housed pigs. Comput Electron Agric 2011;76(2):169–74.

[2] Bengio Y, Nicolas LR, Vincent P, Delalleau O, Marcotte P. Convex neural networks. In: Weiss Y, Schölkopf B, Platt J, editors. Advances in neural information processing systems 18 (NIPS'05), Cambridge: MIT Press; 2005, p. 123–130.

[3] Chen YJ, Jen SL, Li YC, Young MS. The implementation of a high resolution digital color image processing system for animal behavior measurement. In: Proceeding of IEEE EMBS Asian-Pacific Conference on Biomedical Engineering 2003 (IEEE Cat. No.03EX711), Kyoto, Japan; 2003.p. 298–299.

[4] Chen YJ, Li YC, Huang KN, Young MS. The implementation of a stand-alone video tracking and analysis system for animal behavior measurement in morris water maze. In: Proceeding of 27th annual international conference of the IEEE engineering in medicine and biology society (EMBS), Shanghai, China; 2005. p. 1766–1768.

[5] Demšar J. Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 2006;7:1–30.

[6] Friedman N, Russell S. Image segmentation in video sequences: a probabilistic approach. In: Proceeding UAI'97 proceedings of the thirteenth conference on uncertainty in artificial intelligence, providence, Rhode Island, USA; 1997. p. 175–181.

[7] Gederi E, Clifford GD. Fusion of image and signal processing for the detection of obstructive sleep apnea. In: Proceeding of IEEE-EMBS international conference on biomedical and health informatics (BHI), Shenzhen, China; 2012. p. 890–893.

[8] Haque M, Murshed M, Paul M. Improved gaussian mixtures for robust object detection by adaptive multi-background generation. In: Proceeding of19th international conference on pattern recognition, Florida, USA; 2008. p. 1–4.

[9] Hu J, Xin H. Image-processing algorithms for behavior analysis of group-housed pigs. Behav Res Methods Instrum Comput 2000;32(1):72–85.

[10] Huang GB, Zhu QY, Siew CK. Real-time learning capability of neural networks. IEEE Trans Neural Netw 2006;17(4):863–78.

[11] Lin KY, Yang XJ, Wu JH, Chen J, Si HP. A real-time computer vision monitoring way for animal diversity. In: Proceeding of 2012 World Automation Congress (WAC), Puerto Vallarta, Mexico; 2012. p. 1–5.

[12] Kehtarnavaz N, Gamadia M. Real-time image and video processing: from research to reality. In: Bovik AC, editor. Synthesis lectures on image, video, and multimedia processing. Morgan & Claypool Publisher; 2006. p. 1–108.

[13] Lippmann RP. An introduction to computing with neural nets. IEEE ASSP Magazine 1987;4(2):4–22.

[14] Nesterov Y. A method of solving a convex programming problem with convergence rate of (1/k2). In: Proceeding of: soviet mathematics Doklady, 1983;(27):372–376.

[15] Oliviero C, Pastell M, Heinonen M, Heikkonen J, Valros A, Ahokas J, et al. Using movement sensors to detect the onset of farrowing. J Biosyst Eng 2008;100:281–5.

[16] Viazzi S, Borgonovo F, Costa A, Guarino M, Leory T, Berckmans D. Real-time monitoring tool for pig welfare. In: Proceeding of 5th European conference on precision livestock farming, Uppsala, Sweden; 2011. p. 97–104.

[17] Viazzi S, Ismayilova G, Oczak M, Sonoda LT, Fels M, Guarino M, et al. Image feature extraction for classification of aggressive interactions among pigs. Comput Electron Agric 2014;104:57–62.

[18] Zajdel W, Krijnders JD, Andringa T, Gavrila DM. Cassandra: audio-video sensor fusion for aggression detection. In: Proceeding of 2007 IEEE international conference on advanced video and signal based surveillance, London, UK; 2007.p. 200–205.

[19] Zhang WW, Sun J, Tang XO. From tiger to panda: animal head detection. IEEE Trans Image Proc 2011;20(6):1696–708.

[20] Zhou DX, Zhang H. Modified GMM background modeling and optical flow for detection of moving objects. In: Proceeding of 2005 IEEE international conference on systems, man and cybernetics, Waikoloa Village, HI, USA; 2005. p. 2224–2229.

[21] Zivkovic Z. Improved adaptive gaussian mixture model for background subtraction. In: Proceedings of the 17th international conference on pattern recognition. Cambridge, UK; 2004: 2. p. 28–31.