

ANIMAL RECOGNITION SYSTEM BASED ON CONVOLUTIONAL NEURAL NETWORK

*Tibor TRNOVSZKY, Patrik KAMENCAY, Richard ORJESEK,
Miroslav BENCO, Peter SYKORA*

Department of multimedia and information-communication technologies, Faculty of Electrical Engineering,
University of Zilina, Univerzitna 8215/1, 010 26 Zilina, Slovakia

tibor.trnovszky@fel.uniza.sk, patrik.kamencay@fel.uniza.sk, richard.orjeseck@fel.uniza.sk,
miroslav.benco@fel.uniza.sk, peter.sykora@fel.uniza.sk

DOI: 10.15598/aece.v15i3.2202

Abstract. *In this paper, the Convolutional Neural Network (CNN) for the classification of the input animal images is proposed. This method is compared with well-known image recognition methods such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Local Binary Patterns Histograms (LBPH) and Support Vector Machine (SVM). The main goal is to compare the overall recognition accuracy of the PCA, LDA, LBPH and SVM with proposed CNN method. For the experiments, the database of wild animals is created. This database consists of 500 different subjects (5 classes / 100 images for each class). The overall performances were obtained using different number of training images and test images. The experimental results show that the proposed method has a positive effect on overall animal recognition performance and outperforms other examined methods.*

Keywords

Animal recognition system, LBPH, neural networks, PCA, SVM.

1. Introduction

Currently, the animal detection and recognition are still a difficult challenge and there is no unique method that provides a robust and efficient solution to all situations. Generally, the animal detection algorithms implement animal detection as a binary pattern classification task [1]. That means, that given an input image, it is divided in blocks and each block is transformed into a feature. Features from the animal that belongs to a certain class are used to train a certain

classifier. Then, when given a new input image, the classifier will be able to decide if the sample is the animal or not. The animal recognition system can be divided into the following basic applications:

- Identification - compares the given animal image to all the other animals in the database and gives a ranked list of matches (one-to-N matching).
- Verification (authentication) - compares the given animal image and involves confirming or denying the identity of found animal (one-to-one matching).

While verification and identification often share the same classification algorithms, both modes target distinct applications [1]. In order to better understand the animal detection and recognition task and its difficulties, the following factors must be taken into account, because they can cause serious performance degradation in animal detection and recognition systems:

- Illumination and other image acquisition conditions - the input animal image can be affected by factors such as illumination variations, in its source distribution and intensity or camera features such as sensor response and lenses.
- Occlusions - the animal images can be partially occluded by other objects and by other animals.

The outline of this paper is organized as follows. The Sec. 2. gives brief overview of the state-of-the-art in object recognition. In the Sec. 3. , the animal recognition system based on feature extraction and classification is discussed. The obtained experimental results are listed in Sec. 4. Finally, the Sec. 5. concludes and suggests the future work.

2. State of the Art

In [2], an object recognition approach based on CNN is proposed. The proposed RGB-D (combination of a RGB image and its corresponding depth image) architecture for object recognition consists of two separate CNN processing streams, which are consecutively combined with a late fusion network. The CNNs are pre-trained by ImageNet [3]. Depth images are encoded as a rendered RGB images, spreading the information contained in the depth data over all three RGB channels, and then a standard (pre-trained) CNN is used for recognition. Due to lack of large scale labelled depth datasets, CNNs pre-trained on ImageNet [4] are used. A novel data augmentation that aims at improving recognition in noisy real-world setups is proposed. The approach is experimentally evaluated using two datasets: Washington RGB-D Object Dataset and RGB-D Scenes dataset [5].

Another object recognition approach, which uses deep CNN, is proposed in [6]. It also uses CNN, which is pre-trained for image categorization and provide a rich, semantically meaningful feature set. The depth information is incorporated by rendering objects from a canonical perspective and colorizing the depth channel according to distance from the object centre.

3. Animal Recognition System

The image recognition algorithm (image classifier) takes the image (or a patch of the image) as input and outputs what the image contains. In other words, the output is a class label (fox, wolf, bear etc.).

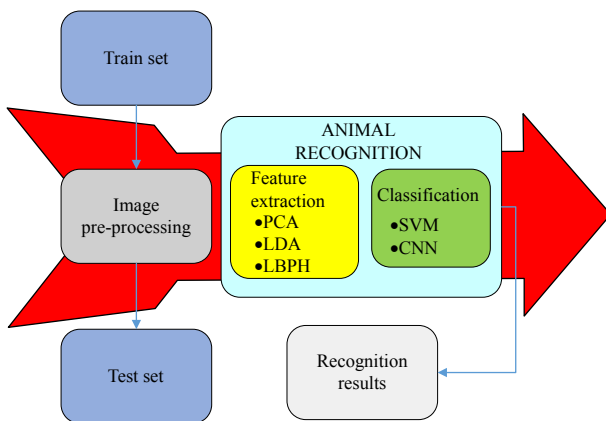


Fig. 1: The animal recognition and classification system.

The animal recognition system (see Fig. 1) is divided into following steps:

- The pre-processing block - the input image can be treated with a series of pre-processing techniques

to minimize the effect of factors that can adversely influence the animal recognition algorithm.

- The feature extraction block - in this step the features used in the recognition phase are computed.
- The learning algorithm (classification) - this algorithm builds predictive model from training data that have features and class labels. These predictive models use the features learnt from the training data on the new (previously unseen) data to estimate their class labels. The output classes are discrete. Types of classification algorithms include decision trees, Support Vector Machines (SVM) and many more.

Interestingly, many traditional computer vision image classification algorithms follow this pipeline (see Fig. 1), while Deep Learning based algorithms bypass the feature extraction step completely.

In all our experiments, the feature extraction (PCA, LDA and LBPH) and classifications (SVM and proposed CNN) methods will be used to estimate test animal images (fox, wolf, bear, hog and deer).

3.1. Principal Component Analysis

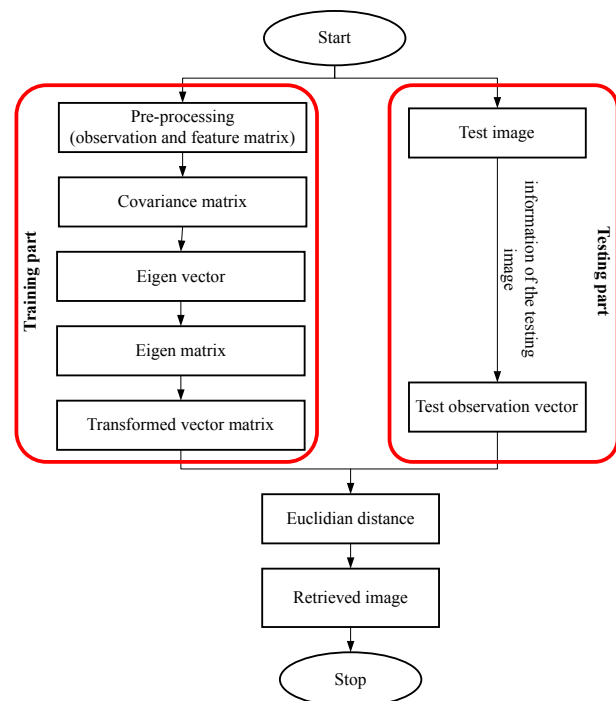


Fig. 2: Block diagram of a PCA algorithm.

The Principal Components Analysis (PCA) is a variable-reduction technique used to emphasize variation and bring out strong patterns in a dataset. The

main idea of the PCA is to reduce a larger set of variables into a smaller set of „artificial“ variables, called „principal components“, which account for most of the variance in the original variables (see Fig. 2) [7] and [8].

The general steps for performing a Principal Component Analysis (PCA):

- Take the whole dataset consisting of d -dimensional samples ignoring the class labels.
- Compute the d -dimensional mean vector (i.e., the means for every dimension of the whole dataset).
- Compute the scatter matrix (alternatively, the covariance matrix) of the whole data set.
- Compute eigenvectors ($e_1, e_2, e_3, e_4, e_5, \dots, e_d$) and corresponding eigenvalues ($\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \dots, \lambda_d$).
- Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a $d \times k$ dimensional matrix \mathbf{W} (where every column represents an eigenvector).
- Use this $d \times k$ eigenvector matrix to transform the samples into the new subspace. This can be summarized by the mathematical equation:

$$y = \mathbf{W}^T \times x, \quad (1)$$

where \vec{x} is a $d \times 1$ dimensional vector representing one sample, and y is the transformed $k \times 1$ dimensional sample in the new subspace.

PCA finds a linear projection of high dimensional data into a lower dimensional subspace such as:

- The variance retained is maximized (maximizes variance of projected data).
- The least square reconstruction error is minimized (minimizes mean squared distance between data point).

3.2. Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is most commonly used as dimensionality reduction technique in the pre-processing step for pattern classification and machine learning applications (see Fig. 3). The goal is to project a dataset into a lower dimensional space with better class separability in order to avoid overfitting and also reduce computational costs [8].

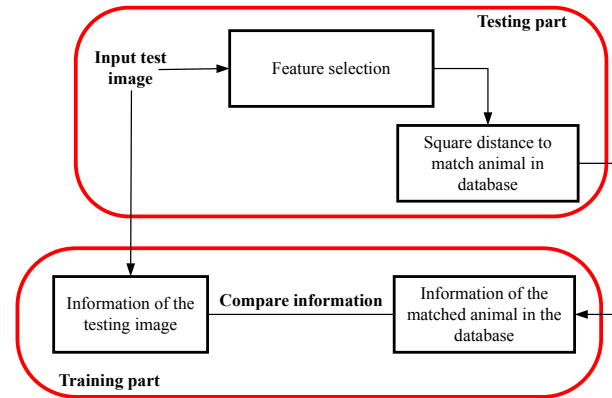


Fig. 3: Block diagram of a LDA algorithm.

The general steps for performing a Linear Discriminant Analysis (LDA) are:

- Compute the d dimensional mean vectors for the different classes from the dataset.
- Compute the scatter matrices.
- Compute the eigenvectors (e_1, e_2, \dots, e_d) and corresponding eigenvalues ($\lambda_1, \lambda_2, \dots, \lambda_d$) for the scatter matrices.
- Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a $d \times k$ dimensional matrix \mathbf{W} (where every column represents an eigenvector).
- Use this $d \times k$ eigenvector matrix to transform the samples into the new subspace. This can be summarized by the matrix multiplication:

$$\mathbf{Y} = \mathbf{X} \times \mathbf{W}, \quad (2)$$

where \mathbf{X} is a $n \times d$ dimensional matrix representing the n samples, and \mathbf{Y} are the transformed $n \times k$ dimensional samples in the new subspace.

The general LDA approach is similar to a Principal Component Analysis (see Fig. 3) [8] and [9].

3.3. LBP Approach to Animal Recognition

The LBPH method takes a different approach than the eigenfaces method (PCA, LDA). In LBPH, each image is analyzed independently, while the eigenfaces method looks at the dataset as a whole.

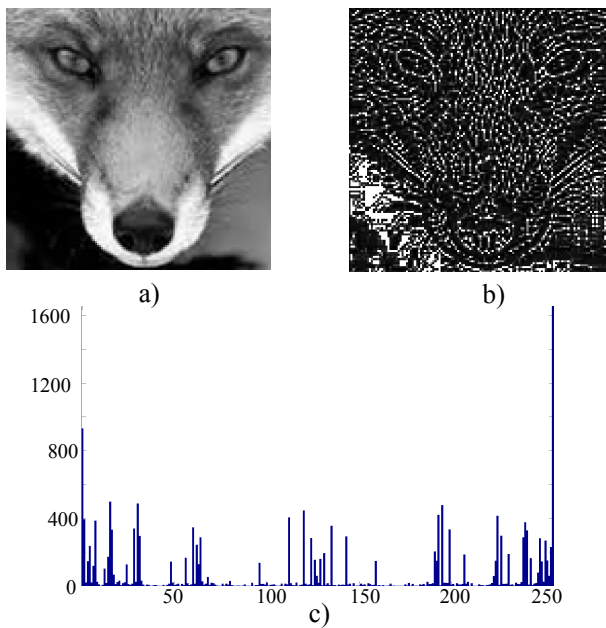


Fig. 4: Local binary patterns of the training dataset: a) input image, b) local binary pattern, c) histogram.

The LBPH method is somewhat simpler, in the sense that we characterize each image in the dataset locally and when a new unknown image is provided, we perform the same analysis on it and compare the result to each of the images in the dataset. The way, which is used for image analysis, does so by characterizing the local patterns in each location of the image. This histogram based approach (see Fig. 4) defines a feature, which is invariant to illumination and contrast [10].

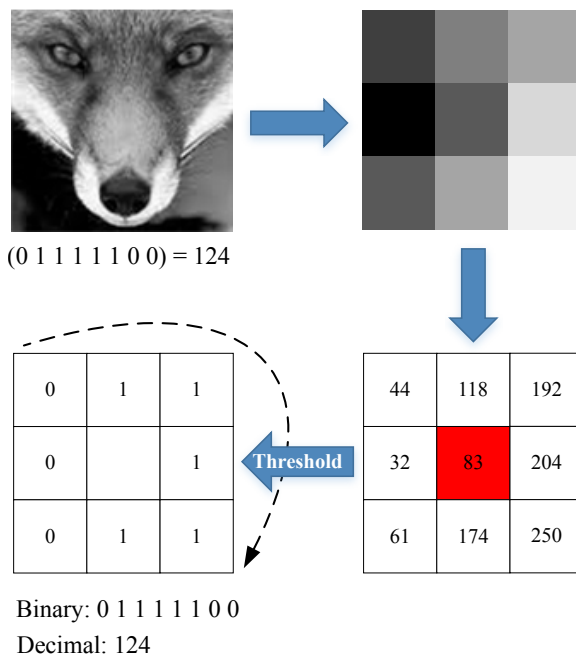


Fig. 5: Example of a LBP calculation (feature extraction).

The basic idea of Local Binary Patterns is to summarize the local structure in a block by comparing each pixel with its neighborhood [10]. Each pixel is coded with a sequence of bits, each of them is associated with the relation between the pixel and one of its neighbors. If the intensity of the center pixel is greater than or equal to its neighbor, then it is denoted with 1. It is denoted 0 if this condition is not met (see Fig. 5). Finally, a binary number (Local Binary Pattern or LBP code) is created for each pixel (just like 01111100). If 8-connectivity is considered, we will end up with 256 combinations [10] and [11]. The LBP operator (used a fixed 3x3 neighbourhood) is shown in Fig. 5.

Training stage (see Fig. 6) looks as follows. Animals and training samples are introduced in the system, and feature vectors are calculated and later concatenated in a unique Enhanced Features Vector to describe each animal image sample. Then, all these results are used to generate a mean value model for each class [11].

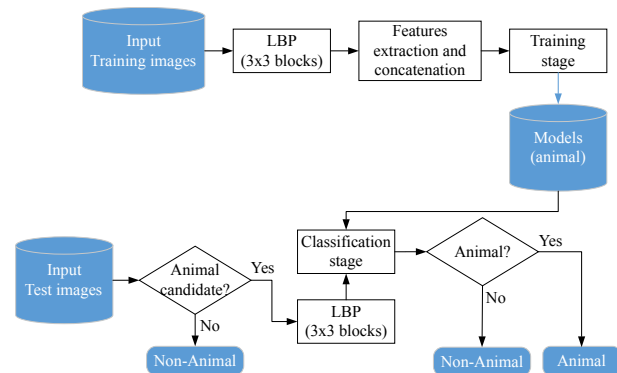


Fig. 6: Block diagram of a LBP algorithm.

Test stage (see Fig. 6) on the other hand looks as follows. For each new test image, segmentation pre-processing is applied first to improve animal detection efficiency. Then the result feeds classification stage. Just test images with positive results in classification stage are classified as animals [11].

3.4. Support Vector Machine

The Support Vector Machine (SVM) is a classification method that samples hyperplanes, which separate two or multiple classes (see Fig. 7). Eventually, the hyperplane with the highest margin is retained, where “margin” is defined as the minimum distance from sample points to the hyperplane. The sample points that form margin are called support vectors and establish the final SVM model [12] and [13].

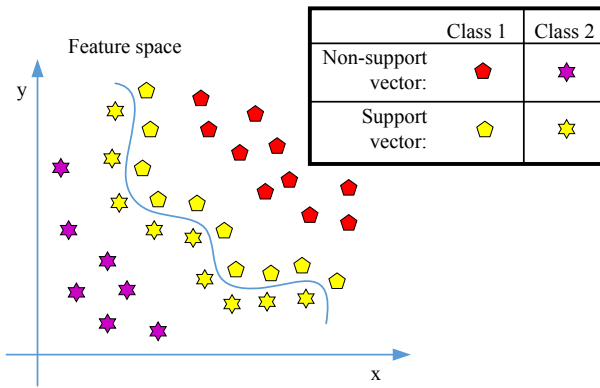


Fig. 7: Boundary searched by the SVM.

Hyper-parameters are the parameters of a classifier that are not directly learned in the learning step from the training data but are optimized separately. The goals of hyper-parameter optimization are to improve the performance of a classifier and to achieve good generalization of a learning algorithm [13].

3.5. Convolutional Neural Network

The Convolutional Neural Networks (CNNs) are a category of Neural Networks that have proven effective in areas such as image recognition and classification. CNN have been successful in identifying animals, faces, objects and traffic signs apart from powering vision in robots and self-driving cars [14].

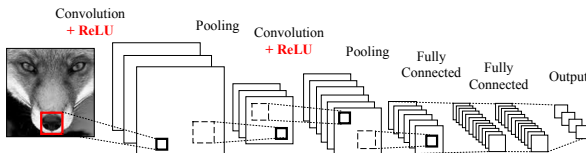


Fig. 8: Example of convolutional neural networks.

The Convolutional Neural Network (see Fig. 8) is similar in architecture to the original LeNet (Convolutional Neural Network in Python) and classifies an input image into categories: fox, wolf, bear, hog or deer (the original LeNet was used mainly for character recognition tasks) [15]. As it is evident from the figure above with a fox image as input, the network correctly assigns the probability for fox among all five categories. There are four main operations in the CNN:

- Convolution.
- Non Linearity (ReLU).
- Pooling or Sub-Sampling (see Fig. 9).
- Classification (Fully Connected Layer).

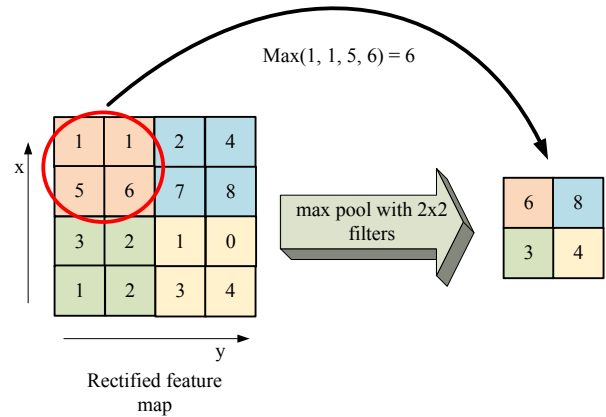


Fig. 9: Max Pooling operation on feature map (2x2 window).

These operations are the basic building blocks of every Convolutional Neural Network, so understanding how these work is an important step to developing a sound understanding of ConvNets [14], [15] and [16].

4. Experiments and Results

In this section, we will evaluate the performance of our proposed method on created animal database. In all our experiments, all animal images were aligned and normalized based on the positions of animal eyes. All tested methods (PCA, LDA, LBPH, SVM and proposed CNN) were implemented in MATLAB and C++/Python programming language.

4.1. Animal Dataset

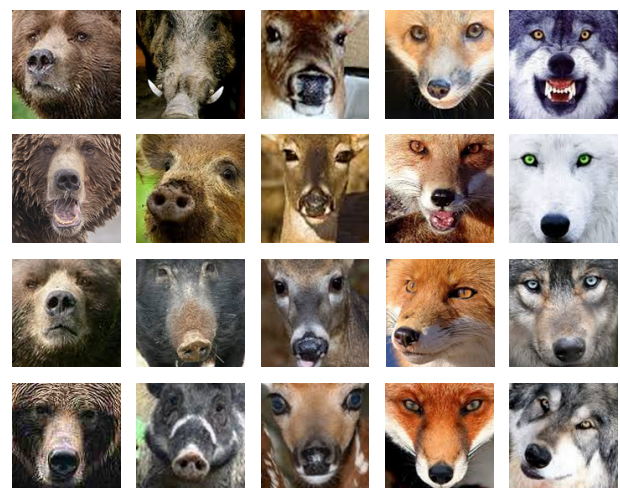


Fig. 10: The example of the created animal database.

The created animal database includes five classes of animals (fox, wolf, bear, hog and deer). Each animal has 100 different images. In total, there are 500 animal

images. The Fig. 10 shows 20 images from the created animal database. The size of each animal image is 150×150 pixels.

There are variations in different illumination conditions. All the images in the created database were taken in the frontal position with tolerance for some side movements. There are also some animal images with variations in scale. The successful animal recognition depends strongly on the quality of the image dataset.

4.2. Experiments

A series of all our experiments for 40, 50, 60, 70, 80 and 90 training images were done. Training database consisted of five classes (fox, wolf, bear, hog and deer). The example of input images from the training database is shown in Fig. 10. All tested methods follow the principle scheme of the image recognition process (see Fig. 1). Training images and test images as a vector were transformed and stored. These images formed the whole created animal database (see Fig. 10). To the designation of feature vector the Euclidean distance was used (accuracy of animal recognition algorithm between the test images and all training images). The obtained results can be seen in Tab. 1.

In order to evaluate the effectiveness of our proposed algorithm we compared the animal recognition rate of our proposed CNN with 4 algorithms (PCA, LDA, LBPH, and SVM). After the system was trained by the training data, the feature space „eigenfaces“ through PCA, the feature space „fisherfaces“ through LDA were found using respective methods. Eigenfaces and Fisherfaces treat the visual features as a vector in a high-dimensional image space. Working with high dimensions was costly and unnecessary in this case, so a lower-dimensional subspace was identified, trying to preserve the useful information. The Eigenfaces method is a holistic approach to face recognition. This approach maximizes the total scatter, but it was a problem in our scenario because the detection algorithm may have generated animal images with high variance due to the lack of supervision in the detection. Although Fisherfaces method can preserve discriminative information with Linear Discriminant Analysis, this assumption basically applies for constrained scenarios. Our detected animal images are not perfect, light and position settings cannot be guaranteed. Unlike Eigenfaces and Fisherfaces, Local Binary Patterns Histograms (LBPH) extract local features of the object and have its roots in 2D texture analysis. The spatial information must be incorporated in the animal recognition model. The proposal in MATLAB is to divide the LBP image into 8×8 local regions using a grid and extract a histogram from each. Then, the spatially en-

hanced feature vector is obtained by concatenating the histograms, not merging them. In our experiments, the SVM classifier used two data types. To create a classification model, training data are used. To test and evaluate trained model accuracy, testing data are used.

The proposal of the Convolutional Neural Network (CNN) is shown in Fig. 11. The input image contains 1024 pixels (32×32 image). The convolutional layer 1 is followed by Pooling Layer 1.

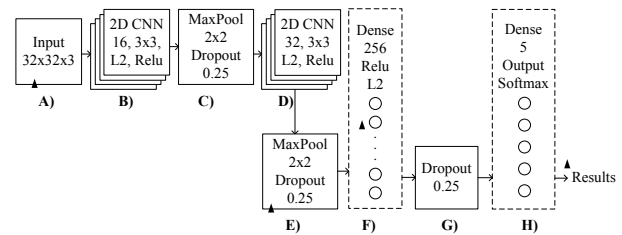


Fig. 11: Block diagram of the proposed CNN.

This convolutional network is divided into 8 blocks:

- A) As input data were used our animal faces from dataset. Each animal face was resized into 32×32 pixels to improve the computation time. The input database has been expanded to provide the better experimental results. This means that the input data were scaled, rotated and shifted.
- B) The second block is 2D CNN layer, which has 16 feature maps with 3×3 kernel dimension. L2 regularization was used due to small dataset. As an activation function, Rectifier linear unit (ReLU) was used.
- C) In this layer the kernel with dimension 2×2 was used and output was dropped out with probability 0.25. It is because we tried to prevent our NN from overfitting.
- D) The second 2D CNN was used with same parameters as first one, but amount of feature maps was doubled to 32.
- E) The MaxPooling layer and Dropout with the same value as in block C were used (see Fig. 11).
- F) As the next layer, standard dense layer was used. It had 256 neurons and as activation function ReLU was used. The L2 regularization was used to better control of weights.
- G) Dropout function was set to 0.25.
- H) As the output dense layer with 5 classes and softmax activation function was used.

In the proposed CNN (see Fig. 11), the pooling operation is applied separately to each feature map. In general, the more convolutional steps we have, the more

complex features (such as edges) it is possible to recognize using proposed network. The whole process is repeated in successive layers until the system can reliably recognize objects. For example, in image classification a CNN may learn to detect edges from raw pixels in the first layer, then use the edges to detect simple shapes in the second layer, and then use these shapes to determine higher-level features, such as facial shapes in higher layers.

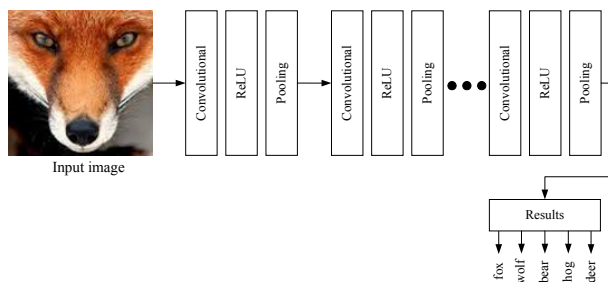


Fig. 12: The example of layers of the proposed CNN.

The neurons in each layer of the CNN (see Fig. 12) are arranged in a 3D manner, transforming a 3D input to a 3D output. For example, for an image input, the first layer (input layer) holds the images as 3D inputs, with the dimensions being height, width, and the colour channels of the image. The neurons in the first convolutional layer connect to the regions of these images and transform them into a 3D output. The hidden units (neurons) in each layer learn nonlinear combinations of the original inputs (feature extraction). These learned features, also known as activations, from one layer become the inputs for the next layer. Finally, the learned features become the inputs to the classifier or the regression function at the end of the network [17].

4.3. Results

The obtained experimental results will be presented in this section. The first row in Tab. 1 presents recognition accuracy using PCA algorithm. The second row (see Tab. 1) presents recognition accuracy using LDA algorithm. The overall accuracy of the LBPH algorithm is described in third row. The experimental results obtained using SVM are described in the next row of the Tab. 1. The last row of the Tab. 1 describes the experimental results of the proposed CNN method (overall recognition accuracy). The all obtained experimental results are divided into six main parts (A, B, C, D, E, and F). The first part of our performed experiments consists of 90 % training images and 10 % test images. The second part consists of 80 % training images and 20 % test images. The third part of our experiments consists of 70 % training images and 30 % test images. The next part consists of 60 % training

images and 40 % test images. The following part consists of 50 % training images and 50 % test images. Finally, the last part of our experiments consists of 40 % training images and 60 % test images.

The ratio of test data and training data (test: training):

- A – 10:90 (90 % of the data was used for training),
- B – 20:80 (80 % of the data was used for training),
- C – 30:70 (70 % of the data was used for training),
- D – 40:60 (60 % of the data was used for training),
- E – 50:50 (50 % of the data was used for training),
- F – 60:40 (40 % of the data was used for training).

The best recognition rate (accuracy of 98 %) using proposed CNN for the first part of our performed experiments (A – 90 % training images and 10 % test images) was achieved. On the other hand, the worst recognition rate (accuracy of 78 %) for the sixth part of our experiments (F – 40 % training images and 60 % test images) was obtained.

Tab. 1: The animal recognition rate for different number of subjects.

| Ratio of test/training animal data | | | | | | |
|------------------------------------|----|----|----|----|----|----|
| | A | B | C | D | E | F |
| PCA (%) | 85 | 77 | 72 | 64 | 62 | 61 |
| LDA (%) | 80 | 70 | 65 | 63 | 61 | 60 |
| LBPH (%) | 88 | 84 | 76 | 73 | 71 | 67 |
| SVM (%) | 83 | 74 | 70 | 68 | 66 | 64 |
| Proposed CNN (%) | 98 | 92 | 90 | 89 | 88 | 78 |

Table 2 displays the confusion matrix for the proposed CNN, constructed using pre-labelled input images from created animal dataset. Using 500 test images, each row corresponds to the image classes (5 classes/100 images for each class), specified by the created animal dataset (target class). The columns indicate the number of times an image, with known image class, was classified as certain class (predicted class).

Tab. 2: The confusion matrix by the proposed CNN method.

| | | Predicted class | | | | | Classification rate |
|--------------|------|-----------------|-----|------|-----|------|---------------------|
| | | bear | hog | deer | fox | wolf | |
| Target class | bear | 97 | 3 | 0 | 0 | 0 | 0.97 |
| | hog | 4 | 91 | 3 | 0 | 2 | 0.91 |
| | deer | 3 | 4 | 93 | 0 | 0 | 0.93 |
| | fox | 0 | 0 | 2 | 95 | 3 | 0.95 |
| | wolf | 0 | 0 | 0 | 5 | 95 | 0.95 |

The cells along the diagonal (green colour) in Tab. 2, represent images which were correctly classified to be

the same class as their pre-labelled image class. Using the correctly classified images, it is possible to determine the classification accuracy. The classification accuracy of the neural network across all classes as the ratio of the sum of the correctly labelled images (green colour) to the total number of images in the test set (500 images) was calculated (accuracy of 94.2 %).

In the Tab. 3, the overall accuracy of correctly identified animals for each class (fox, wolf, bear, hog and deer) using PCA, LDA, LBPH, SVM and proposed CNN is shown.

Tab. 3: The accuracy of correctly identified animals for each class.

| | Bear | Wolf | Fox | Deer | Hog |
|------------------|------|------|-----|------|-----|
| PCA (%) | 82 | 79 | 78 | 76 | 82 |
| LDA (%) | 81 | 77 | 78 | 81 | 83 |
| LBPH (%) | 85 | 87 | 83 | 84 | 82 |
| SVM (%) | 87 | 864 | 85 | 83 | 81 |
| Proposed CNN (%) | 97 | 95 | 95 | 93 | 91 |

The best precision (accuracy of 97 %) using proposed CNN was obtained for the bear class (see Tab. 3). On the other hand, the worst results (accuracy of 76 %) using PCA algorithm was obtained for the deer class.

5. Conclusion

The paper presents a proposed CNN in comparison with the well-known algorithms for the image recognition, feature extraction and image classification (PCA, LDA, SVM and LBPH). The proposed CNN was evaluated on the created animal database. The overall performances were obtained using different number of training images and test images. The experimental result shows that the LBPH algorithm provides better results than PCA, LDA and SVM for large training set. On the other hand, SVM is better than PCA and LDA for small training data set. The best experimental results of animal recognition were obtained using the proposed CNN. The obtained experimental results of the performed experiments show that the proposed CNN gives the best recognition rate for a greater number of input training images (accuracy of about 98 %). When the image is divided into more windows the classification results should be better. On the other hand, the computation complexity will increase.

In the future work, we plan to perform experiments and also tests of more complex algorithms with aim to compare the presented approaches (PCA, LDA, SVM and LBPH) with other existing algorithms (deep learning). We are also planning to investigate reliability of the presented methods by involving larger databases of animal images. Next, we need to improve the perfor-

mance of classifier using combination of local descriptors. Future works can also include experiments with this method on other animal databases.

Acknowledgment

This publication is the result of the project implementation: Centre of excellence for systems and services of intelligent transport, ITMS 26220120028 supported by the Research & Development Operational Programme funded by the ERDF.

References

- [1] XIE, Z., A. SINGH, J. UANG, K. S. NARAYAN and P. ABBEEL. Multimodal blending for high-accuracy instance recognition. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Tokyo: IEEE, 2013, pp. 2214–2221. ISBN 978-1-4673-6356-3. DOI: 10.1109/IROS.2013.6696666.
- [2] EITEL, A., J. T. SPRINGENBERG, L. D. SPINELLO, M. RIEDMILLER and W. BURGARD. Multimodal Deep Learning for Robust RGB-D Object Recognition. In: *2015 IEEE/RSJ, International Conference on Intelligent Robots and Systems (IROS)*. Hamburg: IEEE, 2015, pp. 681–687. ISBN 978-1-4799-9994-1. DOI: 10.1109/IROS.2015.7353446.
- [3] RUSSAKOVSKY, O., J. DENG, H. SU, J. KRAUSE, S. SATHEESH, S. MA, Z. HUANG, A. KARPATY, A. KHOSLA, M. BERNSTEIN, A. C. BERG and L. FEI-FEI. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*. 2015, vol. 115, no. 3, pp. 211–252. ISSN 1573-1405. DOI: 10.1007/s11263-015-0816-y.
- [4] KRIZHEVSKY, A., I. SUTSKEVER and G. E. HINTON. ImageNet classification with deep convolutional neural networks. *Annual Conference on Neural Information Processing Systems (NIPS)*. Harrah's Lake Tahoe: Curran Associates, 2012, pp. 1097–1105. ISBN 978-1627480031.
- [5] RGB-D Object Dataset. In: *University of Washington* [online]. 2014. Available at: <http://rgb-d-dataset.cs.washington.edu/dataset/>.
- [6] SCHWARZ, M., H. SCHULZ and S. BEHNKE. RGB-D object recognition and pose estimation

- based on pre-trained convolutional neural network features. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Seattle: IEEE, 2015, pp. 1329–1335. ISBN 978-1-4799-6923-4. DOI: 10.1109/ICRA.2015.7139363.
- [7] HAWLEY, T., M. G. MADDEN, M. L. O'CONNELL and A. G. RYDER. The effect of principal component analysis on machine learning accuracy with high-dimensional spectral data. *Knowledge Based Systems*. 2006, vol. 19, iss. 5, pp. 363–370. ISSN 0950-7051.
- [8] ABUROMMAN, A. A. and M. B. I. REAZ. Ensemble SVM classifiers based on PCA and LDA for IDS. In: *2016 International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEEES)*. Putrajaya: IEEE, 2016, pp. 95–99. ISBN 978-1-5090-2889-4. DOI: 10.1109/ICAEEES.2016.7888016.
- [9] HAGAR, A. A. M., M. A. M. ALSHEWIMY and M. T. F. SAIDAHMED. A new object recognition framework based on PCA, LDA, and K-NN. In: *11th International Conference on Computer Engineering & Systems (ICCES)*. Cairo: IEEE, 2016, pp. 141–146. ISBN 978-1-5090-3267-9. DOI: 10.1109/ICCES.2016.7821990.
- [10] KAMENCAY, P., T. TRNOVSZKY, M. BENCO, R. HUDEC, P. SYKORA and A. SATNIK. Accurate wild animal recognition using PCA, LDA and LBPH. In: *2016 ELEKTRO*. Strbske Pleso: IEEE, 2016, pp. 62–67. ISBN 978-1-4673-8698-2. DOI: 10.1109/ELEKTRO.2016.7512036.
- [11] STEKAS, N. and D. HEUVEL. Face Recognition Using Local Binary Patterns Histograms (LBPH) on an FPGA-Based System on Chip (SoC). In: *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. Chicago: IEEE, 2016, pp. 300–304. ISBN 978-1-5090-3682-0. DOI: 10.1109/IPDPSW.2016.67.
- [12] FARUQE, M. O. and A. M. HASAN. Face recognition using PCA and SVM. In: *2009 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication*. Hong Kong: IEEE, 2009, pp. 97–101. ISBN 978-1-4244-3883-9. DOI: 10.1109/ICASID.2009.5276938.
- [13] VAPNIK, V. *Statistical Learning Theory*. New York: John Wiley and Sons, 1998. ISBN 978-0-471-03003-4.
- [14] WU, J. L. and W. Y. MA. A Deep Learning Framework for Coreference Resolution Based on Convolutional Neural Network. In: *2017 IEEE 11th International Conference on Semantic Computing (ICSC)*. San Diego: IEEE, 2017, pp. 61–64. ISBN 978-1-5090-4284-5. DOI: 10.1109/ICSC.2017.57.
- [15] LeNet - Convolutional Neural Networks in Python. In: *PyImageSearch* [online]. 2016. Available at: <http://www.pyimagesearch.com/2016/08/01/lenet-convolutional-neural-network-in-python/>.
- [16] Understanding convolutional neural networks. In: *WildML* [online]. 2015. Available at: <http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/>.
- [17] MURPHY, K. P. *Machine Learning: A Probabilistic Perspective*. Massachusetts: MIT University Press. 2012. ISBN 9780262018029.

About Authors

Tibor TRNOVSZKY was born in Zilina and started to study on University of Zilina in 2009. He is focused on computer vision and image processing.

Patrik KAMENCAY was born in Topolcany in 1985, Slovakia. He received his M.Sc. and Ph.D. degrees in Telecommunications from the University of Zilina, Slovakia, in 2009 and 2012, respectively. His Ph.D. research work was oriented to reconstruction of 3D images from stereo pictures. Since October 2012 he is researcher at the Department of MICT, University of Zilina. His research interest includes holography for 3D display and construction of 3D objects of the real scene (3D reconstruction).

Miroslav BENCO was born in Vranov nad Toplou in 1981, Slovakia. He received his M.Sc. degree in 2005 at the Department of Control and Information System and Ph.D. degree in 2009 at the Department of Telecommunications and Multimedia, University of Zilina. Since January 2009 he is researcher at the Department of MICT, University of Zilina. His research interest includes digital image processing.

Peter SYKORA was born in Cadca in 1987, Slovakia. He received his M.Sc. degree in Telecommunications from the University of Zilina, Slovakia, in 2011. His research interest includes hand gesture recognition, work with 3D data, object classification, machine learning algorithms.