Model of Graphic Object Identification in a Video Surveillance System based on a Neural Network

Andii Sahun[1], Vladyslav Khaidurov[2], and Viktor Bobkov[2]

[1] *National University of Life and Environmental Sciences of Ukraine, 15 Heroyiv Oborony str., Kyiv, 03041, Ukraine*
[2] *National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute," 37 Peremohy ave., Solomyanskyi district, Kyiv, 03056, Ukraine*

Abstract

The object identification system, given the correct model selection and settings, enables accurate and fast identification of graphical objects in video data. A deep learning neural network is the base for the identification system. The use of the CamVid benchmark video dataset for training the neural network model allows using of fundamental truth labels that associate each pixel with one of the 32 semantic classes of the identification system. The total number of used training images is 421, and the testing ones are 280. Selecting optimal parameters for the learning function and identification support, a method for measuring the distance between feature vectors gives the necessary result—the identification of objects from the video data stream of perimeter IP cameras demonstrated an average accuracy of 99.7% across all cameras in the test examples, consisting of 12 video fragments with a duration of 70 seconds each. The developed algorithm of the system is capable of identifying objects of 11 classes from the graphical information content of IP cameras.

Keywords

Deep machine learning, neural network, classifier, clustering, cluster analysis, pattern recognition.

1. Introduction

The creation and implementation of object identification systems within various technical systems is explained by their demand in a large number of contemporary information systems. Today, effective and high-quality identification of various objects can only be achieved by intelligent systems. These systems are based on artificial intelligence or machine learning algorithms. Frequently, neural networks in various variations serve as the basis for object identification systems. When using neural networks, it is crucial to correctly determine both the type of network and choose a test video database for proper training of the neural network [1, 2]. Otherwise, the accuracy of the classification in the created object identification system may be low. Another important aspect is the formation of the classifier's gradation scale in the identification system.

If we need to analyze video data it is critical to create a function that analyzes video frames and utilizes test databases for initial object identification in individual static frames of the video sequence [3]. A separate challenge in object identification models based on neural networks is the classification of recognized objects [4].

Significant contributions to creating algorithms and methods for object identification have been made by: V. Kornienko, L. Budkova, A. Korobov, A. Korobov [1], V. Lakhno, V. Chubaievskyi, K. Palaguta [6], V. Kornienko, I. Gulina, L. Budkova [7], O. Kryvoruchko, A. Desiatko, A. Blozva, V. Semidotska [8]. Publications on the use of neural networks and machine learning

CPITS-2024: Cybersecurity Providing in Information and Telecommunication Systems, February 28, 2024, Kyiv, Ukraine
EMAIL: avd29@ukr.net (A. Sahun); allif0111@gmail.com (V. Khaidurov); vbb.wlp@gmail.com (V. Bobkov)
ORCID: 0000-0002-5151-9203 (A. Sahun); 0000-0002-4805-8880 (V. Khaidurov); 0000-0002-1567-8186 (V. Bobkov)

technologies for solving applied tasks in object identification and cybersecurity are dedicated to the research of the following scholars: S. Schuster [9], L. Ljung, C. Andersson, K. Tiels [10], O. Nelles [11], I. Goodfellow, Y. Bengio, A. Courville [12], C.-J. Lin [13], T. Schön [14], D. Kandamali, X. Cao, M. Tian, Z. Jin, H. Dong, K. Yu [15], S. Bickler [16], B. Akhmetov [7], and others.

In most cases, overcoming the aforementioned challenges allows obtaining a correct mathematical model of the object identification system. The same systems can be used as a computer vision system or for other practical applications. To obtain an object identification system, it is crucial to obtain a mathematical model on which such a system will operate [17]. To achieve this, it is necessary to analyze existing algorithms, models, and methods applied in intelligent object identification systems.

2. Graphic Object Identification

According to the task conditions, the fundamental property of the identification system is the need to distinguish and identify objects in individual frames of video content. It is rational to base such models on the mathematical framework and algorithms of neural networks.

The advantage of a mathematical model based on neural networks is the ability to learn. In computer intelligent systems for object identification, machine learning is a factor that significantly improves the adequacy and accuracy of recognition and identification algorithm performance. The most rational type of learning for the mathematical model of the created system is supervised learning. This type of learning involves using labeled or selected datasets that contain input data and expected output results. Thus, during the learning process, the model can perform internal iterations to approximate the specified error boundary. Once the learning boundary (error) ceases to exceed the specified limits, the model is considered trained.

Deep learning allows training a model to predict outcomes based on a set of input data. For network training, both supervised and unsupervised learning can be used. The difference between these two learning methods is illustrated in Figure 1.

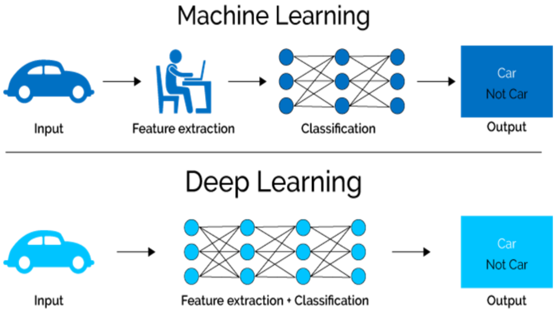

Figure 1: The difference between two learning methods (Machine Learning and Deep Learning)

In the neural network used as the basis for the object identification system, multiple layers of neurons are envisaged. The input layer of the neural network takes initial input data. In this case, there are four neurons in the output layer: intensity of each pixel, Haar features for each of the considered objects (trees, cars, roads, sky, pedestrians, etc.). The input layer passes this data to the first hidden layer. Hidden layers perform mathematical computations with the input data. The term "depth" in "deep learning" refers to having more than one hidden layer. The output layer produces the final result. In this case, it is the identification of the type of object present in a specific image. The diagram of the obtained neural network model is shown in Figure 2.

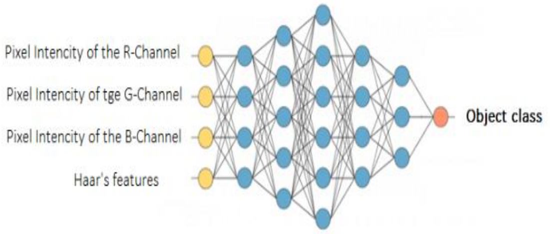
Pixel Intencity of the R-Channel
Pixel Intencity of tge G-Channel
Pixel Intencity of the B-Channel
Haar's features
Object class

Figure 2: The best forecasting models obtained by the method of group accounting of arguments obtained

Identification of objects based on this machine learning method occurs through the reevaluation of the weights of connections between neurons. The weight factor determines the importance of the input data element. When classifying objects, the weight coefficient in interneuronal connections is the most crucial.

For the practical implementation of the created model, the vgg16() function in MATLAB was chosen as the basis. This function represents the architecture of a deep neural network for image classification. It includes 16 convolutional and fully connected layers.

The given function has 16 layers, including 13 convolutional layers and 3 fully connected layers. The overall architecture of VGG-16 includes:

1. Input Layer: Consisting of 224x224 pixels.
2. Convolutional Layers: each convolutional layer has 3x3 filters and ReLU activation.

The number of filters in each convolutional layer increases from 64 to 512.

MaxPooling (2x2) layers are used after each block of convolutional layers.

3. Fully Connected Layers: three fully connected layers with 4096 neurons each. ReLU activation is applied to the output neurons of each fully connected layer. Dropout (random deactivation) may be applied to prevent overfitting.
4. Output Layer: the output layer has 1000 neurons (the number of classes in ImageNet images). The softmax activation function is used to obtain class probabilities.

The vgg16 function in MatLab returns a neural network object but does not include a specific method for measuring the distance between feature vectors.

There are some specific methods for measuring the distance between feature vectors: Euclidean distance; Euclidean distance squared; Manhattan distance; Chebyshev distance; and hamming distance [18]. But Euclid's distance has the biggest advantage—its simplicity. Its calculation is the simplest and light-calculated way to get a direct path between two points. In the task of graphic image identification, we use the Euclidean distance to compare the model's output values with expected values during training and for classifying objects based on their features.

In this research, the classical Euclidean distance is utilized. To calculate it, we use the following expression:

$$\rho(x,x') = \sqrt{\sum_{i=1}^{n}\left(x_i - x_i'\right)^2},$$

where x_i is the first n-dimensional vector, x_i' is the second n-dimensional vector.

We can define these two vectors as follows: $x = (x_1; x_2; \dots x_n)$. $x' = (x_1'; x_2'; \dots x_n')$.

To train a neural network, prepared data needs to be fed into it, and the generated outputs should be compared with the results from a test dataset. For the training of the neural network, test examples from the video database labeled Cambridge (CamVid) were used. This database represents the first collection of videos with semantic labels of object classes, complemented with metadata. The database provides fundamental ground truth labels associating each pixel with one of 32 semantic classes. It was obtained from a free internet link.

This database addresses the need for forming experimental data to quantitatively evaluate identification and classification algorithms. For each pair of objects, the "distance" between them is measured, representing the degree of similarity.

A model that provides a minimum of external criteria is considered optimal. With an increase in the number of variables in the model and the degree of the reference polynomial, obtaining the best forecasting model can increase significantly.

The obtained model is practically implemented in the MatLab environment. The aforementioned video database of test samples was used for training the neural network. This video database contains a 10-minute frame with a rate of 30 Hz. Images are segmented using corresponding semantic labels at a frequency of 1 Hz and partially at 15 GHz. The CamVid database has four datasets corresponding to the studied objects. They include:

1. Pixel-level semantic segmentation for more than 700 images (segmentation performed manually), later verified and confirmed by a second person for accuracy.
2. High-quality color video images in high resolution, collected in the database. These images represent digital video recordings with a long duration of video content.
3. Contain calibrated sequences for color response and internal camera characteristics, considering the point of view, as a typical surveillance camera and fixation of each frame in the sequences.

4. To support the expansion of the database, software is proposed for labeling (necessary to assist users who want to perform accurate labeling of classes for other images and videos).

The relevance of the database is evaluated by measuring the algorithm’s performance in each of the three different areas: object recognition in multiple classes, pedestrian detection, and label propagation.

3. Training and Testing the Identification System Model

To ensure the operational functionality of the model, the following steps were taken: loading test datasets from the CamVid database sets and preparing a repository for test loading samples.

Declare individual classes of identified objects: “Sky,” “Building,” “Pole,” “Road,” “Pavement,” “Tree,” “SignSymbol,” “Fence,” “Car,” “Pedestrian,” “Bicyclist”.

The resolution of training frames is set to 360×480 points of the video stream: imageSize = [360 480 3].

A neural network model for the identification of graphical objects returns a specific set of numerical values for the identified objects. To obtain the central value of an ordered set of such data, we use the median () function in the Matlab environment. The initialization of parameters for neural network training and the definition of the error function for the neural network are provided in Table 1.

Table 1
Parameters (argument) of the neural network training function

Argument’s Name	Value
Momentum	0.9
InitialLearnRate	1×10^{-3}
L2Regularisation	5×10^{-4}

As we see from Table 1, the trainingOptions() function has 9 arguments:

- sgdm (Stochastic Gradient Descent with Momentum).
- 'Momentum (momentum helps accelerate the optimization process by incorporating information from previous iterations).
- InitialLearnRate (initial neural network learning rate).
- L2Regularization (L2 neural network training regularization—weight decay);
- MaxEpochs (number of full passes through the entire dataset during network training).
- MiniBatchSize (mini-batch size, the number of examples used to update the gradient at each iteration).
- Shuffle and every epoch (shuffling data at each epoch during training).
- VerboseFrequency (the frequency of displaying training progress information in the command window).

Further, it is necessary to define the classifier’s grading scales to perform cluster analysis and the final classification of identified graphical objects. As reference data, we will take the informational component of the color channels of the identified image {R ∈ (0; 255), G ∈ (0; 255), B ∈ (0; 255)} this way:

- Reference information vector for sky identification:
  [128 128 128; ... % “Sky”].
- Reference information vector for building and structure identification:
  000 128 064; ... % “Bridge”
  128 000 000; ... % “Building”
  064 192 000; ... % “Wall”
  064 000 064; ... % “Tunnel”
  192 000 128; ... % “Archway”.
- Reference information vector for identifying columns, pillars, etc:
  192 192 128; ... % “Column_Pole”
  000 000 064; ... % “TrafficCone”.
- Reference information vector for identifying road surface:
  128 064 128; ... % “Road”
  128 000 192; ... % “LaneMkgsDriv”
  192 000 064; ... % “LaneMkgsNonDriv”.
- Reference information vector for identifying sidewalks, pavement, and pedestrian paths:
  000 000 192; ... % “Sidewalk”
  064 192 128; ... % “ParkingBlock”
  128 128 192; ... % “RoadShoulder”.
- Reference information vector for identifying trees, shrubs, and other significant vegetation areas:
  128 128 000; ... % “Tree”
  192 192 000; ... % “VegetationMisc”.

- Reference information vector for identifying road signs, informational signs, etc.:
  192 128 128; ... % “SignSymbol”
  128 128 064; ... % “Misc_Text”
  000 064 064; ... % “TrafficLight”.
- Reference information vector for identifying fences and barriers:
  064 064 128; ... % “Fence”.
- Reference information vector for identifying vehicles:
  064 000 128; ... % “Car”
  064 128 192; ... % “SUVPickupTruck”
  192 128 192; ... % “Truck_Bus”
  192 064 128; ... % “Train”
  128 064 064; ... % “OtherMoving”.
- Reference information vector for identifying pedestrians, animals, and light means of manual cargo transportation:
  064 064 000; ... % “Pedestrian”
  192 128 064; ... % “Child”
  064 000 192; ... % “CartLuggagePram”
  064 128 064; ... % “Animal”.
- Reference information vector for identifying light mechanized means of transportation for people and cargo (motorcycles/bicycles):
  000 128 192; ... % “Bicyclist”
  192 000 192; ... % “MotorcycleScooter”.

From the provided reference information vectors, it can be seen that some classes of the identified and subsequently clustered data contain subclasses, namely:

- The ‘Building’ class contains 4 embedded subclasses.
- The ‘Pole’ class contains 2 subclasses.
- The ‘Road’ contains 3 subclasses.
- The ‘Pavement’ contains 3 subclasses.
- The ‘Tree’ contains 2 subclasses.
- The ‘SignSymbol’ contains 3 subclasses.
- The ‘Car’ contains 5 subclasses.
- The ‘Pedestrian’ contains 4 subclasses.
- The ‘Bicyclist’ contains 2 subclasses.

The color channel reference map for highlighting the classes of identified objects contains a color channel vector for 11 basic classes. Its representation is shown in Table 2. It is critical to define a reference map for color channels.

When training the model, the total number of training images is 421, and the number of test images is 280. As a result of training, the weight of each class can be determined (Table 3, column ‘Class weight’).

Table 2
Basic classes of model

Class name	Class characteristics (RGB features)
Sky	128 128 128
Building	128 0 0
Pole	192 192 192
Road	128 64 128
Pavement	60 40 222
Tree	128 128 0
SignSymbol	192 128 128
Fence	64 64 128
Car	64 0 128
Pedestrian	64 64 0
Bicyclist	0 128 192

Table 3
The weight of each class

Class name	Class weight	IoU
Sky	0.318184709354742	0.9266
Building	0.208197860785155	0.7987
Pole	5.092367332938507	0.1698
Road	0.174381825257403	0.9518
Pavement	0.710338097812948	0.4188
Tree	0.417518560687874	0.4340
SignSymbol	4.537074815482926	0.3251
Fence	1.838648261914560	0.4920
Car	1.000000000000000	0.0688
Pedestrian	6.605878573155874	0
Bicyclist	5.113338416059593	0

Frequency characteristics of the occurrence of weights for individual classes on specific frames are also determined in column ‘IoU’ of Table 3. As shown in column ‘IoU’, there are any pedestrians and Bicyclist in the test image were identified.

Through training on the training set, the algorithm based on a deep learning neural network distinguishes the background from the informational content of identification (object—car) (Figure 3).

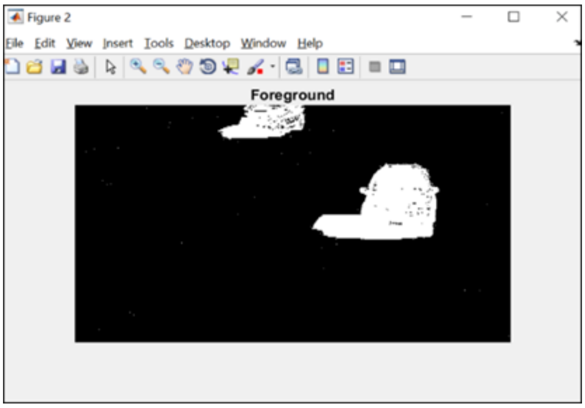

Figure 3: The algorithm of the created identification model separates the background in a graphical video frame

As an example, to showcase the operation of the developed object identification system, a graphical frame depicted in Figure 4 has been loaded.

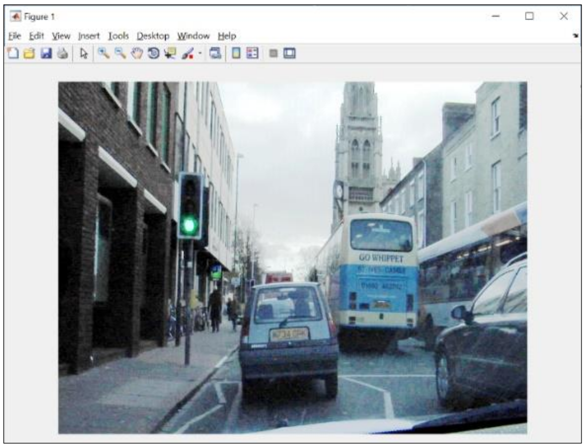

Figure 4: Original graphical frame for testing the identification model

As a result of the developed algorithm, we observe the identification results on the demonstration test frame of objects subjected to further cluster analysis. The legend for the identification algorithm classes is provided in Figure 5.

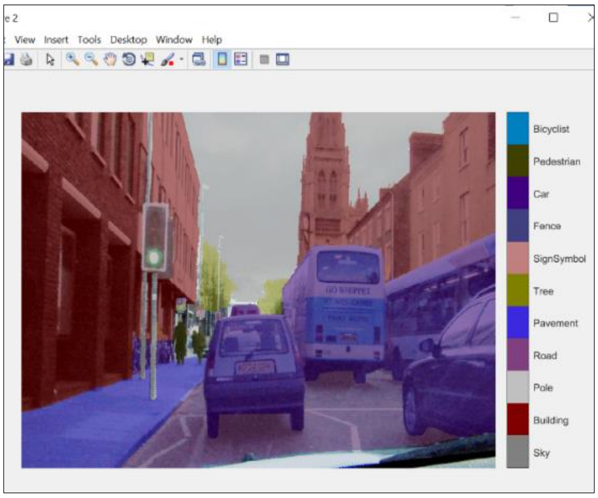

Figure 5: The legend for the identification algorithm classes

The practical application of the developed identification system implies that the identification results were presented not in a visual (graphic) form but in the form of a numerical data array, allowing these results to be further used in more complex systems. To achieve this, we construct a histogram of the frequency of occurrence of identified classes and subclasses of objects in the identification zone of video surveillance cameras (Figure 6).

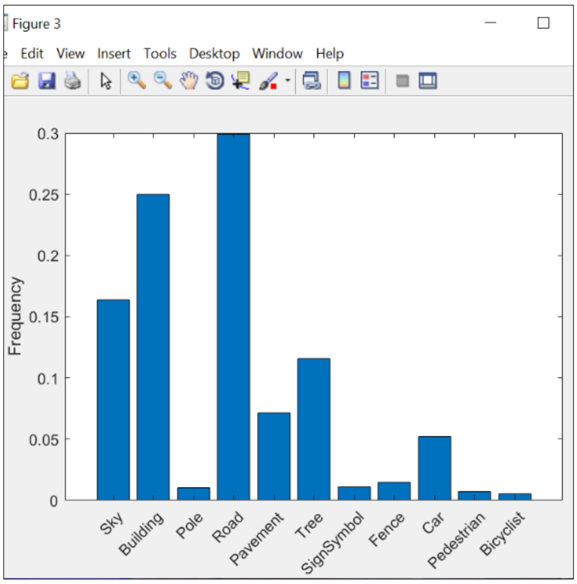

Figure 6: A histogram of the frequency of occurrence of identified classes and subclasses of objects in the identification zone of video surveillance cameras

By comparing deviations of the obtained numerical arrays with reference ones, the developed system for identifying graphic objects can make decisions regarding the detection of specific incidents or determine certain reactions of the system.

4. Conclusions

As a result of testing the implemented algorithm in the Matlab environment, the accuracy of the graphic information identification system model was proven to be high. The identification of objects from the video data stream of perimeter surveillance cameras demonstrated an average accuracy of 96.38% across all cameras in the test examples, consisting of 12 video fragments with a duration of 68.35 seconds each. These results were achieved due to several factors, including:

- Type of neural network and thoughtfully chosen parameters of the neural network training function.
- Method for determining the similarity measure of an object to existing classes (distance measure for clustering) through Euclidean distance.
- Using the Cambridge (CamVid) labeled video database as a collection of videos with semantic labels of object classes, accompanied by metadata. This

database facilitated effective training of the foundation of the identification system—the neural network model.

- Corrected definition of the classifier's grading scales to perform cluster analysis and the final classification of identified graphical objects and others.

As seen from the results of the image identification model, incorporating Haar's features into the neural network model yields excellent results in the classification and identification of images of various types.

References

[1] B. Bebeshko, et al., Application of Game Theory, Fuzzy Logic and Neural Networks for Assessing Risks and Forecasting Rates of Digital Currency, J. Theor. Appl. Inf. Technol. 100(24) (2022) 7390–7404.

[2] K. Khorolska, et al., Application of a Convolutional Neural Network with a Module of Elementary Graphic Primitive Classifiers in the Problems of Recognition of Drawing Documentation and Transformation of 2D to 3D Models, J. Theor. Appl. Inf. Technol. 100(24) (2022) 7426–7437.

[3] V. Sokolov, P. Skladannyi, A. Platonenko, Video Channel Suppression Method of Unmanned Aerial Vehicles, in: IEEE 41st International Conf. on Electronics and Nanotechnology (2022) 473–477. doi: 10.1109/ELNANO54667.2022.9927105.

[4] V. Zhebka, et al., Optimization of Machine Learning Method to Improve the Management Efficiency of Heterogeneous Telecommunication Network, in: Cybersecurity Providing in Inf. and Telecom. Syst., vol. 3288 (2022) 149–155.

[5] V. Moskalenko, A. Korobov, Optimization Parameters of Intellectual Identification System of Objects on the Terrain, Radioelectron. Comput. Syst. 2 (2019) 32–39. doi: 10.32620/reks.2016. 2.05.

[6] V. Lakhno, et al., Information Security Audit Method Based on the Use of a Neuro-Fuzzy System, Software Engineering Application in Informatics, LNNS 232 (2021) 171–184. doi: 10.1007/978-3-030-90318-3_17.

[7] O. Herasina, V. Korniienko, The Algorithms of Global and Local Optimization in Tasks of Identification of Difficult Dynamic Systems, Inf. Processing Syst. 6(87) (2010) 73–77.

[8] V. Lakhno, et al., Development Strategy Model of the Informational Management Logistic System of a Commercial Enterprise by Neural Network Apparatus, in: Cybersecurity Providing in Inf. and Telecom. Syst., vol. 2746 (2020) 87–98.

[9] H. Schuster, Deterministic Chaos: Introduction and Recent Results, Nonlinear Dyn. Solids (1992) 22–30. doi: 10.1007/978-3-642-95650-8_2.

[10] L. Ljung, et al., (2020). Deep Learning and System Identification, IFAC-PapersOnLine 53(2) (2020) 1175–1181. doi: 10.1016/j.ifacol.2020.12.1329.

[11] O. Nelles, Nonlinear System Identification: From Classical Approaches to Neural and Fuzzy Models, Springer (2001). doi: 10.1007/978-3-662-04323-3.

[12] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, The MIT Press (2016).

[13] C.-J. Lin, SISO Nonlinear System Identification Using a Fuzzy-Neural Hybrid System, Int. J. Neural Syst. 08(03) (1997) 325–337. doi: 10.1142/ s0129065797000331.

[14] C. Andersson, N. Wahlström, T. Schön, Learning Deep Autoregressive Models for Hierarchical Data, IFAC-PapersOnLine 54(7) (2021) 529–534. doi: 10.1016/j.ifacol.2021.08.414.

[15] D. Kandamali, et al., Machine Learning Methods for Identification and Classification of Events in ϕ-OTDR Systems: a review, Appl. Opt. 61(11) (2022) 2975. doi: 10.1364/ao.444811.

[16] S. Bickler, (2018). Machine Learning Identification and Classification of Historic Ceramics. Archaeology in New Zealand 61 (2018) 48–58.

[17] V. Buriachok, et al., Invasion Detection Model using Two-Stage Criterion of Detection of Network Anomalies, in: Cybersecurity Providing in Inf. and Telecom. Syst., vol. 2746 (2020) 23–32.

[18] J.-H. Lee, Minimum Euclidean Distance Evaluation Using Deep Neural Networks, AEU – Int. J. Electron. Commun. 112 (2019) 152964. doi: 10.1016/j.aeue. 2019.152964.