International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V69I10P219

Volume 69 Issue 10, 151-160, October, 2021

Original Article

Neural Networks As A Tool For Pattern Recognition of Fasteners

Yasser Mohammad Al-Sharo[1], Amer Tahseen Abu-Jassar[2], Svitlana Sotnik[3], Vyacheslav Lyashenko[4]

[1]*Faculty of Computer Science and Information Technology, Ajloun National University, Jordan*
[2]*Faculty of Computer Science and Information Technology, Ajloun National University, Jordan*
[3]*Department of Computer-Integrated Technologies, Automation, and Mechatronics, Kharkiv National University of Radio Electronics, Ukraine*
[4]*Department of Media Systems and Technology, Kharkiv National University of Radio Electronics, Ukraine*

[4]lyashenko.vyacheslav@gmail.com

Abstract *–The work is devoted to the study of pattern recognition features of industrial parts in individual fasteners' forms. The main types of neural network architectures and their features are considered. Neural networks are classified into separate categories for ease of perception and analysis. An approach to recognition of hardware products such as fasteners using neural network, which is implemented in Python using Keras machine learning library, is proposed. The main generators are described: for training data, testing, and validation. Codes fragments of corresponding programs for implementation of the proposed approach to pattern recognition of fasteners are presented.*

Keywords – *Neural Networks, Recognition, Fastener, Hardware, Program Code.*

I. INTRODUCTION

Neural networks (NN) are a universal "tool" that can be used both for solving simple problems and for solving problems that require complex analytical calculations[1]-[3].

Neural networks are actively expanding their field of application, and process automation has become the norm in the industry, where cheap and readily available automation technologies have become the standard option available to manufacturing companies [4]. However, without realizing flexibility in production through the use of intelligent systems, the use of technology will be limited. Therefore, networks are used to enhance production processes. For example, NN is used to solve problems of monitoring the technological process of manufacturing precision engineering parts. Or, as for example of NN active use, detection of a defect in steel and its subsequent classification is one of the urgent tasks in the development of monitoring and quality control systems in production.

Thus, using neural networks, you can implement classification, prediction, or, for example, recognition. The latter direction is most relevant recently in terms of NN use. Moreover, among tasks of recognition can be:

- face recognition and other biometric data;
- voice recognition;
- text recognition;
- barcode recognition;
- license plate recognition;
- recognition of technical objects.

There are many different methods of pattern recognition, ranging from classical algorithms that compare various parameters of an image with a sample (often these parameters are specified as vectors found by certain functions) to machine learning algorithms (Deep Learning) that use neural networks for classification and characterization images.

In modern enterprises, along with the identification of personnel, recognition of technical objects is increasingly used for automatic analysis and verification of production facilities for compliance with certain requirements. Recognition can be on a conveyor line, in warehouses, or in general, searching for a specific screw or bolt in the huge catalog of the online store using one photo. For example, in the latter case, the problem is that today there are thousands of parts models. Each detail has its own description and characteristics, so there is no hope for filters; that is, the task of recognizing details is the priority and is completely unsolved because the variety and complexity of recognition tasks do not make it possible to implement one universal approach to the solution.

Recognition of the same parts in various processes allows you to achieve maximum production results.

In this work, emphasis is on metal fasteners, since today it is impossible to make complex and reliable structures without the use of metal products; one of the most common types is hardware (HW).

II. MATERIALS AND METHODS

A. Related work

Questions concerning the use of neural networks for building image recognition systems are considered by many authors.

Currently, pattern recognition systems have become widespread in various spheres of human activity [5]-[9].

In [7], authors describe recognition in meteorology in terms of individual watersheds' topography for predicting the amount of precipitation.

The use of neural networks in medicine for automatic detection of pulmonary nodules on CT images using deep convolutional neural networks is highlighted in [8]. The authors use a Faster R-CNN structure with two networks.

Or, for example, recognition in the transport industry [9] was implemented on the basis of a deep neural network – road signs recognition. The authors carried out an analysis of spatial transformers and methods of stochastic optimization. The work was reduced to the development of a convolutional neural network, which should improve the modern problem of classifying road signs.

In [10], pattern recognition in horticulture is described. Since an important mechanism of plant immunity is based on the recognition of conservative microbial molecules, work is about determining the degree of plant necrosis and apple tree's resistance to bacterial burns.

The demand for object recognition in production has also led to the emergence of a large number of relevant studies [4], [11]-[13].

In [11], recognition of objects in industrial production to control automatic production of goods. The authors considered deep learning of specially designed convolutional neural networks for defect detection and recognition of industrial objects (mechanical parts of technological systems or machines). Six publicly available industrial datasets were examined containing defective materials and industrial tools or engine parts. Following the recent success of the Virtual Geometry Group (VGG) network, authors have proposed a modified version called Multipath VGG19 that allows more local and global functionality to be retrieved.

The work [12] is aimed at studying action recognition based on deep learning in an industrial environment. A feature of work is that authors propose a method that combines multiple deep learning networks, including CNNs, spatial transformer networks (STNs), and convolutional graph networks (GCNs), to process video data in industrial workflows. The proposed method extracts both spatial and temporal information from video data.

The recognition of parts catalog objects for efficient control of drawings distribution is described in [13]. Here, to solve the recognition problem, authors combine CNN (Convolutional Neural Network) and LSTM (Long Short Term Memory). Extracts key points via CNN to separate part catalog letters or numbers and leader lines or part features, and when using LSTM to recognize strings of letters or numbers.

The solution of object recognition problems through assessment of posture and shape is well presented by authors in [14]. An overview of many existing advances in the field of finding the pose of an object based on shape, appearance, based on characteristics, and comparison of their accuracy, complexity, and performance is given.

The basic principles and methods of image recognition are presented in [15]. The paper describes the composition of the image recognition system.

The basic principles of CNN are studied in [16]. This is where authors carry out face recognition.

The works [17]-[19] are devoted to issues of building and choosing the architecture of the neural network.

In [17], the design of convolutional neural network (CNN) architectures are shown. The authors introduced MetaQNN, a reinforcement learning-based meta-modeling algorithm for automatically generating high-performance CNN architectures.

An overview of deep neural networks architectures and their applications is presented in [18]. The application of deep learning methods in some selected areas (speech recognition, pattern recognition, and computer vision) is considered.

The architectures of parallel recurrent neural networks are described in [19]. The p-RNN architectures for modeling sessions based on clicks and functions (images and text) of selected elements are considered.

Also, the architecture of artificial neural networks and learning processes are described in [20].

Design patterns for deep convolutional neural networks [21].

The basics of designing a recognition system are described in [22].

B. Features of neural network architecture

Recognition of any patterns using a neural network is impossible without an algorithm for creating the neural network, which must be adapted to solve tasks and usually consists of 3 main stages: choice of network architecture, network training, network application.NN allows the process of large amounts of statistical data at high speed and predicts output parameters of the network with a high degree of probability, taking into account risk assessment and optimization of main resource flows. The components of the neural network include neurons – models representing threshold value, and connections between neurons – synapses. At the same time, there are many features of NN, ranging from structure and ending with the nature of training. Fig. 1 summarizes and presents the main classification features of NN architecture.

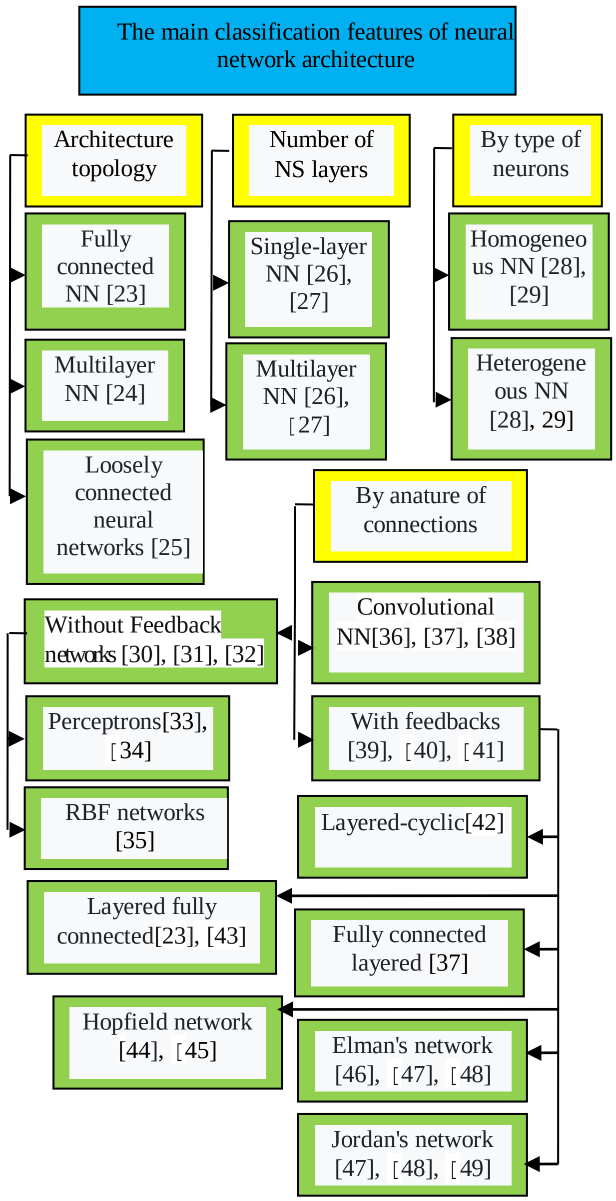

Fig. 1: Classification features of NN architecture

The main differences, features, directions of using the architecture of neural networks for individual classification criteria are summarized and presented in corresponding tables 1-5.

Table 1: The main features and differences of the architecture of neural network according to topological basis

Features Fully connected neural networks [23]	Features Multi-layered ANNs [24]	Features Loosely connected networks [25]
- fully-connected layers require a large amount of memory, which makes it difficult to further increase the size of the model; - such network makes it possible to implement the linear mapping between input and output directly; - networks are easy to parallelize, and, as a result, to reduce training time; - learns very quickly initially, as weight gradient is instantly available thanks to the direct connection of inputs to exits; - for correct data training, it is required to increase the number of hidden neurons, which leads to an increase in the number of parameters, and, as a consequence, reduces learning rate, requires large training sample; - inferior in accuracy to networks with convolutional layers; - data classification during recognition; - are used more often for character recognition.	- to simulate functions of almost any degree of complexity; - identification of dynamic systems and their control; - training of such neural network implies solving the problem of stochastic approximation of mapping of input signals of a neural network to output, often without extracting information about nature and properties of mapping itself; - for construction of complex multidimensional areas based on a large number of training examples, allows you to use them as a classifier for pattern recognition; - to solve forecasting problems.	- in loosely bound networks lack parallel paths between vertices; - principle of genetic similarity of structure is implemented; - simple implementation of regular structures of any dimension, suitable for implementation in large-scale integrated circuit technology; - for processing high-dimensional data; - ability to recognize high-dimensional patterns.

The number of neurons and layers must be selected based on the complexity of the problem amount of analyzed data and resulting classes, as well as available computing resources.

Thus, multilayer neural networks have much greater capabilities than single-layer neural networks.

Table 2: Some features of NN by number of layers

Features Single-layer neural network	Features Multilayer neural network
- possibility of optimal learning both without teacher and with the teacher; - for dynamic identification and management of systems; - image coding and texture segmentation; - for character recognition (both printed and handwritten); - to identify linear systems; - to classify images; - to approximate functions; - for prediction.	- training of such neural network implies solving the problem of stochastic mapping approximation of input signals of a neural network to output, often without extracting information about nature and properties of mapping itself; - to classify images; - for solving problems of forecasting by time series; - for prediction and control; - for character recognition (both printed and handwritten); - for speech recognition; - face recognition and other biometric data; - barcode recognition; - license plate recognition; - recognition of technical objects.

Table 3: The main directions of using NN architecture without feedbacks

Perceptrons	RBF networks
- to solve forecasting problems; - for pattern recognition; - to classify images; - for character recognition; - for recognizing audio signals (for example, for recognizing genres of music compositions); - to recognize tactical situations (for example, in robotics).	- to classify images; - to approximate functions (excellent approximators in the field of data change); - for prediction and control; - for data forecasting.

Without Feedback, networks are widely used to solve a class of problems such as forecasting, clustering, and recognition. Among multilayer networks without feedbacks, the distinction is also made between fully connected (output of each neuron of the q-th layer is connected to the input of each neuron of (q + 1)-th layer) and partially fully connected.

Compared to multilayer perceptrons, RBF networks have the ability to learn more quickly; however, they require long preparation and setup time due to the need to perform more complex calculations.

A convolutional neural network is a form of a multi-layer neural network without feedbacks. It is the main tool for classifying and recognizing objects. A convolutional neural network allows you to simultaneously reduce the amount of information stored in memory, due to which it copes better with higher-resolution pictures, and to highlight reference features of image, for example, edges, contours, or faces.

Table 4: The main directions of using NN architecture based on Hopfield, Elman, and Jordan networks

Hopfield network [41]	Elman's network [42]-[44]	Jordan's network [43], [44]
- teaching associative memory; - for associative memory; - to solve combinatorial optimization problems.	- in control systems for moving objects to detect changes in signal characteristics; - to solve problems of forecasting by time series (even on highly noisy time series); - for pattern recognition; - to classify images.	- for pattern recognition; - to classify images; - solves the same class of problems as the Elman network but has better approximating and predictive properties due to deeper memory and an additional layer of nonlinear activation functions.

Feedback networks contain loops in their structure, which ensures the influence of output signal on classification process in future. This enables multiple participation of neurons in the processing of input data and reduces the volume of the network through the use of feedbacks. Layered-cyclic – layers are closed in the ring: the last layer transmits its output signals to the first one [42]. Layered fully connected – consist of layers, each of which is a fully connected network, and signals are transmitted both from layer to layer and within layer [23], [43]. Fully connected layer – they do not separate phases of exchange within the layer and transfer to the next one [37].

The most famous representatives of recurrent neural networks are Hopfield's network, Jordan's network, and Elman's network [47], [48].

The main model used in tasks of image recognition and analysis is a convolutional neural network, less often for audio. The success of this model is largely due to its ability to take into account the two-dimensional topology of the image, in contrast to the multilayer perceptron.

The main model used in speech recognition problems is considered to be deep neural networks and recurrent neural networks.

Networks without feedbacks under the condition of "learning with teacher" are mainly used for approximation of functions, classification. Feedback networks subject to "supervised learning" are mainly used for time series forecasting, online learning. Networks without feedbacks are simpler to implement than recurrent networks. A neural network without feedback is characterized by a number of layers and their constituent neurons. There is no rule for determining these parameters. The more neurons and layers, the greater network capabilities, nonlinearity of the relationship between input and output increases, but the learning rate decreases.

Table 5: Some features of NN by nature of connections

Without Feedback networks (direct distribution)	Convolutional neural network	Feedback networks (Recurrent networks)
- with a large number of classes and a large number of inputs, training feedforward network takes a lot of time and resources; - possibility of optimal learning both without teacher and with the teacher; - for speech recognition; - classification of images; - for texture recognition; - face recognition (for example, facial expressions); - to predict (for example, chemical shifts of carbon); - for dynamic identification and management of systems.	- when recognizing patterns, they are used because they imply the presence of a large number of different classes of patterns; - convolutional networks are invariant to shifts and distortions of the input signal; - character recognition; - great efficiency of the convolutional network in recognizing handwritten numbers; - speech recognition; - facerecognitioninphotographs, etc.	- classification of images; - for speech recognition; - for text recognition (including recognition of non-segmented continuous handwritten text); - for processing sequentially ordered data that do not have simple temporal interpretation (for example, chemical structures that are represented as trees).

Unsupervised learning networks without feedback are mainly used for data compression and feature extraction. Feedback networks under the condition of "unsupervised learning" are mainly used for associative memory, data clustering, optimization.

Let us also highlight the following distinctive features of NN:

- number of neurons in each layer can be any and in no way connected in advance with the number of neurons in other layers.

- key property of neural networks is their ability to learn, which makes neural network models indispensable for solving problems for which algorithmicization is impossible, problematic or too laborious.

- you should not choose complex neural network for processing big data, you should transform data using standard algorithms for existing solutions.

Then, when choosing neural network architecture, it is necessary to take into account:

- type of problem being solved (approximation, forecasting, clustering);

- type of input data (dimensions, key factors, that is, some architectures require fine tuning of several parameters, etc.).

III. RECOGNITION OF HARDWARE PRODUCTS SUCH AS FASTENERS USING NEURAL NETWORK

In this work, recognition task will consist in comparing characteristics of hardware with previously known ones and assigning hardware to one of classes (that is, implementation of classification).

Thus, task of neural network will be to classify set of images of hardware images.

Photos of various hardware will be set as set of images.

The developed system will distribute given set of images into 2 classes (photos of bolts and screws are given to general resolution and for ease of entry into database, it was decided to enumerate such images with continuous numbering from 1 to 500), in fig. 2 and fig. 3 shows some of these photos:

- bolts (fig. 2);

- screws (fig. 3).

Fig. 2: Examples of images in "Bolts" class

Usually, to classify objects in image means to indicate number (or name of class) to which object belongs, depending on its vector of features. Then rules for correlating an image to one of classes are called classifier. In this case, corresponding label is attached to each image of hardware, describing class to which each of images belongs.

Fig. 3: Examples of images in "Screws" class

During preliminary testing, these images, after preprocessing, will be fed to inputs of neural network. At end of each such testing of NN, preliminary assessment of network performance on test set is presented. This check makes it possible to assess effectiveness of network in process of training network.

When hardware recognition system developing, following tasks will be implemented:

- import of set of images;

- preprocessing (preprocessing) images;

- transmission of obtained images to inputs of neural system;

- analysis of results obtained.

1.Using camera, take series of images containing recognition objects and import set of images by categories.

The images are randomly distributed into 3 classes: training, validation and test.

2. To reliably determine characteristic features of hardware images, it is necessary to process original images and bring them to certain form – bringing all images to single format:

- conversion to single binary format;

- conversion to single image resolution;

- conversion to single color format (RGB).

The process of image preprocessing is an obligatory stage in hardware recognition, since thanks to preprocessing it is possible to improve accuracy of selected characteristic features of hardware images.

For example, by taking series of shots of various options for hardware: with rotation around central axis; "Blurry" pictures; "Cropped" pictures, etc. several models are trained; learning to recognize each hardware, even if it is not so located (rotated) or if distortion.

During evaluation, new objects are processed by randomly selected model, thus complicating creation of “learner” model.

For high defensive effectiveness during training, it is necessary to choose different models.

3. Resulting images are transmitted to inputs of neural system.

In this work, we will implement backpropagation convolutional neural network that predicts boundaries of object at each position.

The first layer of this network will be convolutional layer (fig. 4), where y1 is bolt; y2 – screw.

The next layer of this model will be activation layer. The most commonly used activation function in pattern recognition is Rectified Linear Unit (ReLU).

Then pooling layer is added to reduce computational complexity.

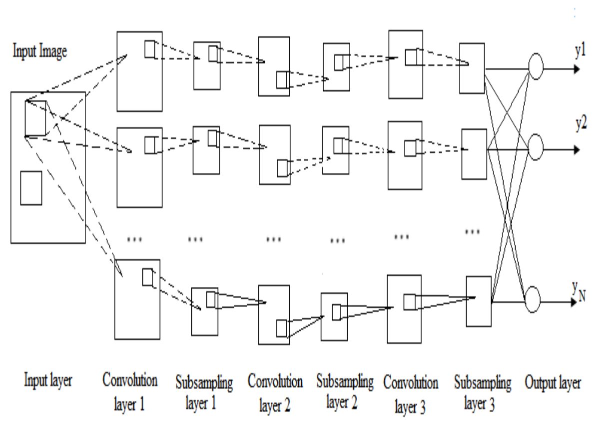

Fig. 4: General structure of convolutional network

Thus, there is already convolutional block consisting of above functions. This block will be repeated 3 times in succession. The next step is to create fully connected layers. In this case, only 2 fully connected layers will be described: one with ReLU activation function, second output, associated with sigmoidal function used to assess accuracy of obtained neural network.

An example of convolution and downsampling operation is shown in fig. 5.

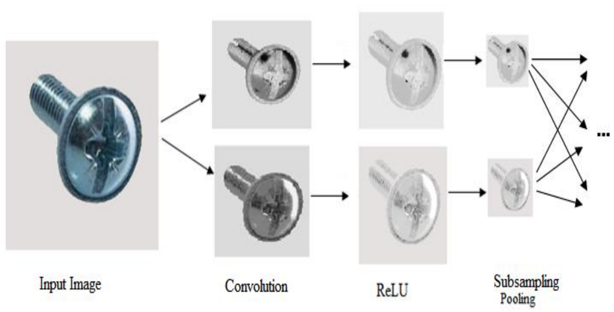

Fig. 5: Example of convolution and downsampling operation (subsampling)

Convolutional Network (CNN) is main tool for classifying and recognizing objects in photographs [36]-[38], [50].

There are many applications for CNN, such as Deep Convolutional Neural Network (DCNN), Region-CNN (R-CNN), Fully Convolutional Neural Networks (FCNN), Mask R-CNN and others [38], [51]-[56].

In backpropagation, kernels must learn weights to generate features from local set of inputs only.

Convolutional neural networks, unlike other neural network architectures, provide partial resistance to scale changes, displacements, rotations, camera angles, and other distortions.

So, first layer of neural network will be convolutional layer.

When passing through first layer, various distinctive features of image are enhanced, most often boundaries of objects. This transformation is performed using filter – array of given dimension.

4. Analysis of results obtained.

The recognition accuracy over entire test set can be determined by expression:

$$R = n / N,$$

where N is number of elements in test samples;

n is number of correctly recognized patterns from test sample.

As implementation language, development language was chosen – Python.

When recognizing and classifying images, most common are TensorFlow and Keras.

Keras was chosen in work because it is high-level API that allows you to implement many powerful, but often complex TensorFlow functions as simply as possible, moreover, it is configured to work with Python without any major changes or settings.

In Keras, loading dataset is very easy, and images themselves need only minimal preprocessing and at same time very high speed, second only to Tensorflow.

Also,Python interpreter with development environment (IDE), set of Keras dependency libraries, and PlaidML framework were used, which will accelerate training of neural network using resources of video card.

PlaidML can use AMD graphics card for deep learning on Windows platform, which is relevant for this study.

The first step in writing program is to import libraries:

```
importplaidml.keras
plaidml.keras.install_backend()

importkeras
fromkeras
import layers
importos
```

The ImageDataGenerator class is very useful for classifying images. That is, ImageDataGenerator can be used to expand, train or test datasets, or to create dataset from existing images, so dataset is created next. A snippet of code for defining image import class is shown below:

```
datagen =
keras.preprocessing.image.ImageDataGenerator(
rescale=1. / 255,
horizontal_flip=True,
shear_range=0.2,
rotation_range=40,
zoom_range=0.2)
```

The rescale parameter sets values of each pixel to range (0, 1).

The shear_range, rotation_range and zoom_range, horizontal_flip parameters randomly transform images within specified limits by several parameters: blur, rotation and scaling and mirroring.

Three generators were used: train_generator – for training data, valid_generator – validation, and testing_generator – testing. Application snippets are given below:

```
train_generator = datagen.flow_from_directory(
// Training set
'data/training',
target_size=(180, 180),
batch_size=64,
class_mode='categorical',
color_mode="rgb",
shuffle=True)

valid_generator = datagen.flow_from_directory(
// Validation set
'data/validation',
target_size=(180, 180),
batch_size=16,
class_mode='categorical',
color_mode="rgb",
shuffle=True)
testing_generator = datagen.flow_from_directory(
// Testing set
'data/testing',
target_size=(180, 180),
batch_size= 50,
class_mode='categorical',
color_mode="rgb",
shuffle=False)
```

So, target_size argument interpolates all images up to 180x180 resolution.

color_mode converts color space to RGB format.

Batch_size determines number of items loaded at time into RAM.

Class_mode sets mode for setting image labels, and shuffle shuffles them randomly.

Next, we build network. To create your own data generator, you need to inherit from Sequence class. Here model is set asobject of keras.Sequential () class, after which layers of neural network are added using keras.Sequential.add () method. Snippet of code:

```
model = keras.Sequential()
model.add(layers.Conv2D(32, (3, 3),
input_shape=(180, 180, 3)))
model.add(layers.Activation('relu'))
model.add(layers.MaxPooling2D(pool_size=(2, 2)))
```

Each layer is defined by specific class of keras.layers module.

layers.Conv2D () – Specifies 2D convolutional layer to which images will be fed.

layers.Activation () – sets activation layer (ReLU in this case).

layers.MaxPooling2D () – downsampling layer that selects maximum element in each cell of dimension (2, 2).

When the model is loaded, it does not contain any information about weight values. Then, before starting testing, you need to train network.

The keras.Sequential.fit_generator () method was used to train network. Code snippet:

```
model.fit_generator(generator=train_generator,
steps_per_epoch=32,
epochs=500,
validation_data=valid_generator,
validation_steps=16)
```

After training neural network, accuracy of its work was assessed. Snippet of code:

```
model.evaluate(testing_generator, steps=2)
```

Time spent on learning the network: 1 hour 41 minutes.

The result of evaluating accuracy of network is presented in table. 6.

Table 6: Assessment of object recognition accuracy

Evaluated Image Classes	Accuracy	Loss function estimation
Bolts	0.87	1.3126
Screws	0.80	0.7934

The data were obtained taking into account number of elements in test samples equal to 250 images.

IV. CONCLUSION

In this work, method using neural networks was chosen for pattern recognition, since such networks have fairly high accuracy even with relatively simple network structure.

The study of features of neural network architectures types is carried out, which is one of stages on way to creating an optimal recognition system, as a result of analysis, four categories of neural networks are identified and features of each architecture are describe

To carry out recognition, class of objects was chosen – hardware.

To implement recognition of hardware images, database of two classes elements images (screws and bolts) was created on basis of neural network.

The recognition implementation was in Python using Keras machine learning library. It was Keras libraries that were chosen, since these libraries allow both loading data and evaluating effectiveness of model.

To implement system, PlaidML framework was used, which made it possible to accelerate training of neural network.

Thus, an example of implementation of system for recognizing hardware products such as fasteners using neural network is proposed

The developed system makes it possible to recognize type of details in image.

The accuracy of network is 80 % (tested on 500 images, 250 images for each class.

REFERENCES

[1] P. Orobinskyi, D. Petrenko, V. Lyashenko, Novel Approach to Computer-Aided Detection of Lung Nodules of Difficult Location with Use of Multifactorial Models and Deep Neural Networks, In 2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM). (2019) 1-5.

[2] R. Matarneh, S. Maksymova, Zh. Deineko, V. Lyashenko, Building Robot Voice Control Training Methodology Using Artificial Neural Net, International Journal of Civil Engineering and Technology. 8(10) (2017) 523-532.

[3] M. Ayaz, T. Sinelnikova, S. K. Mustafa, V. Lyashenko, Features of the Construction and Control of the Navigation System of a Mobile Robot, International Journal of Emerging Trends in Engineering Research. 8(4) (2020) 1445-1449.

[4] A. D O'Riordan, D. Toal, T. Newe, G Dooly, Object recognition within smart manufacturing, Procedia Manufacturing, 38 (2019) 408-414.

[5] V. Lyashenko, F. Laariedh, S. Sotnik, M. A. Ahmad, Recognition of Voice Commands Based on Neural Network,TEM Journal. 10(2) (2021) 583-591.

[6] J. H. Baker, V. Lyashenko, S. Sotnik, F. Laariedh, S.K. Mustafa, M. A. Ahmad, Some Interesting Features of Semantic Model in Robotic Science, International Journal of Engineering Trends and Technology. 69(7) (2021) 38-44.

[7] F. R. Lin, N. J. Wu, T. K. Tsay, Applications of cluster analysis and pattern recognition for typhoon hourly rainfall forecast, Advances in Meteorology. 2017 (2017) Article ID 5019646.

[8] H. Xie, D. Yang, N. Sun, Z. Chen, Y. Zhang, Automated pulmonary nodule detection in CT images using deep convolutional neural networks, Pattern Recognition. 85 (2019) 109-119.

[9] Á. Arcos-García, J. A. Alvarez-Garcia, L. M. Soria-Morillo, Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods, Neural Networks. 99 (2018) 158-165.

[10] S. Piazza, et al., The Arabidopsis pattern recognition receptor EFR enhances fire blight resistance in apple, bioRxiv. (2021) doi:10.1101/2021.01.22.427734.

[11] J. Li, et al., Training convolutional neural networks with synthesized data for object recognition in industrial manufacturing, In 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). (2019) 1544-1547.

[12] Z. Jiao, G. Jia, Y. Cai, Ensuring Computers Understand Manual Operations in Production: Deep-Learning-Based Action Recognition in Industrial Workflows, Applied Sciences. 10(3) (2020) 966.

[13] S. W. Choi, C. S. Song, C. S. Kim, Parts Catalog Object Recognition Technology for Efficient Drawing Distribution Management, In 2019 IEEE International Conference on Big Data (Big Data). (2019) 6022-6023.

[14] A. A. Fathima, Object recognition through pose and shape estimation. arXiv e-prints, (2020) arXiv:2006.12864.

[15] Y. Chen, X. Zhou, Research and Implementation of Robot Path Planning Based on Computer Image Recognition Technology, In Journal of Physics: Conference Series. 1744(2) (2021) 022097.

[16] J. Wang, Z. Li, Research on face recognition based on CNN. In IOP Conference Series: Earth and Environmental Science, 170(3) (2018) 032110.

[17] B. Baker, et al., Designing neural network architectures using reinforcement learning, arXiv preprint. (2016) arXiv:1611.02167.

[18] W. Liu, et al., A survey of deep neural network architectures and their applications, Neurocomputing. 234 (2017) 11-26.

[19] B. Hidasi, M. Quadrana, A. Karatzoglou, D. Tikk, Parallel recurrent neural network architectures for feature-rich session-based recommendations, In Proceedings of the 10th ACM conference on recommender systems. (2016) 241-248.

[20] I. N. Da Silva, et al. Artificial neural network architectures and training processes, Artificial neural networks. (2017) 21-28.

[21] L. N. Smith, N. Topin, Deep convolutional neural network design patterns, arXiv preprint. (2016) arXiv:1611.00847.

[22] S. Samarasinghe, Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition, Crc Press. (2016).

[23] S. Albawi, T. A. Mohammed, S. Al-Zawi, Understanding of a convolutional neural network, In 2017 International Conference on Engineering and Technology (ICET). (2017) 1-6.

[24] P. M. Nguyen, Mean field limit of the learning dynamics of multilayer neural networks, arXiv preprint. (2019) arXiv:1902.02880.

[25] R. Wu, R. Srikant, J. Ni, Learning loosely connected Markov random fields, Stochastic Systems. 3(2) (2013) 362-404.

[26] A. G. Salman, et al., Single layer & multi-layer long short-term memory (LSTM) model with intermediate variables for weather forecasting, Procedia Computer Science. 135 (2018) 89-98.

[27] B. M. Wilamowski, Neural network architectures, Intelligent Systems. (2018) 6-17.

[28] T. Cohen, M. Geiger, M.Weiler, A general theory of equivariantcnns on homogeneous spaces,arXiv preprint. (2018) arXiv:1811.02017.

[29] V. Christou, et al., Hybrid extreme learning machine approach for heterogeneous neural networks,Neurocomputing, 361 (2019) 137-150.
[30] A. Haldorai, A. Ramu, Canonical Correlation Analysis Based Hyper Basis Feedforward Neural Network Classification for Urban Sustainability, Neural Processing Letters. (2020) 1-17.
[31] F. Han, et al., A survey on metaheuristic optimization for random single-hidden layer feedforward neural network, Neurocomputing. 335 (2019) 261-273.
[32] S. Dutta, et al., Output range analysis for deep feedforward neural networks, NASA Formal Methods Symposium. (2018) 121-138.
[33] G. Li, et al., Fast learning network with parallel layer perceptrons, Neural Processing Letters. 47(2) (2018) 549-564.
[34] S. Ravanbakhsh, Universal equivariant multilayer perceptrons, International Conference on Machine Learning (PMLR). (2020) 7996-8006.
[35] P. H. Zadeh, R. Hosseini, S. Sra, Deep-rbf networks revisited: Robust classification with rejection, arXiv preprint. (2018) arXiv:1812.03190.
[36] L. Wen, et al., A new convolutional neural network-based data-driven fault diagnosis method, IEEE Transactions on Industrial Electronics. 65(7) (2017) 5990-5998.
[37] K. Chellapilla, S. Puri, P. Simard, High performance convolutional neural networks for document processing, In Tenth international workshop on frontiers in handwriting recognition. (2006) inria-00112631.
[38] G. Yao, T. Lei, J. Zhong, A review of convolutional-neural-network-based action recognition, Pattern Recognition Letters. 118 (2019) 14-22.
[39] A. R. Zamir, et al., Feedback networks, Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 1308-1317.
[40] C. Jarvers, H. Neumann, Incorporating feedback in convolutional neural networks, Proceedings of the Cognitive Computational Neuroscience Conference. (2019) 395-398.
[41] J. Gough, M. R. James, The series product and its application to quantum feedforward and feedback networks, IEEE Transactions on Automatic Control. 54(11) (2009) 2530-2544.
[42] A. Antakli, et al., Intelligent distributed human motion simulation in human-robot collaboration environments, Proceedings of the 18th International Conference on Intelligent Virtual Agents. (2018) 319-324.
[43] E. Domany, R. Meir, Layered neural networks, Models of neural networks. (1991) 307-334.
[44] Y. Yu, et al., A review of recurrent neural networks: LSTM cells and network architectures, Neural computation. 31(7) (2019) 1235-1270.
[45] C. Yin, et al., A deep learning approach for intrusion detection using recurrent neural networks, Ieee Access. 5 (2017) 21954-21961.
[46] H. Ramsauer, et al., Hopfield networks is all you need,arXiv preprint.(2020) arXiv:2008.02217.
[47] K. Kolanowski, et al.,Multisensor data fusion using Elman neural networks, Applied Mathematics and Computation. 319 (2018) 236-244.
[48] W. Wu, et al., Time series analysis of human brucellosis in mainland China by using Elman and Jordan recurrent neural networks, BMC infectious diseases. 19(1) (2019) 1-11.
[49] S. Kumari, et al., Convolutional elmanjordan neural network for reconstruction and classification using attention window, Innovations in Computational Intelligence and Computer Vision. (2021) 173-181.
[50] Y. Zhang, et al., An unsupervised parameter learning model for RVFL neural network, Neural Networks. 112 (2019) 85-97.
[51] J. Zhang, K. Shao, X. Luo, Small sample image recognition using improved Convolutional Neural Network, Journal of Visual Communication and Image Representation. 55 (2018) 640-647.
[52] F. Fang, et al., Combining faster R-CNN and model-driven clustering for elongated object detection, IEEE Transactions on Image Processing. 29 (2019) 2052-2065.
[53] M. F. Spadea, et al., Deep convolution neural network (DCNN) multiplane approach to synthetic CT generation from MR images-application in brain proton therapy, International Journal of Radiation Oncology Biology Physics. 105(3) (2019) 495-503.
[54] J. H. Baker, et al., Some Interesting Features of Semantic Model in Robotic Science, International Journal of Engineering Trends and Technology. 69(7) (2021) 38-44.
[55] A. Ma.Babker, et al., Information technologies of the processing of the spaces of the states of a complex biophysical object in the intellectual medical system health, International Journal of Advanced Trends in Computer Science and Engineering. 8(6) (2018) 3221-3227.
[56] H. Pratt, et al., Fcnn: Fourier convolutional neural networks, Joint European Conference on Machine Learning and Knowledge Discovery in Databases. (2017) 786-798.