

Application of Neural Network Technologies in the Tasks of Quality Control of Textile Products

Kaznacheeva A.A. ^a, Vlasenko O.M. ^b, Zakharkina S.V. ^c, Ryzhkova E.A. ^d
and Stupak M.M. ^e

Russian State University named after A.N. Kosygin (Technologies. Design. Art), Moscow, Russia

Keywords: Machine learning, Neural network, Defect images, Textile material, Training, Testing, Accuracy.

Abstract: The article sets the task of developing an intelligent automated system for detecting defects in textile materials. The analysis of machine learning and deep learning algorithms is carried out in relation to solving the problem of product quality control. The implementation of an artificial neural network implemented in a Raspberry Pi microcontroller and receiving a set of input data in the form of a large stream of images from a high-speed digital camera is considered. The stages of creating a model in the Python high-level programming language in the Google Colab cloud development environment using TensorFlow and Keras libraries for deep learning are described. The development process includes the preparation of initial data intended for training and testing the system, as well as checking the operation of the resulting neural network, which consists in recognizing images of defects on the tissue according to classification criteria.

1 INTRODUCTION

Creating an automated quality control system for textile products based on an intelligent approach is a complex and time-consuming task. The solution of this problem is possible today by means of artificial intelligence, using machine vision technologies, machine learning algorithms (Machine learning) and deep learning – artificial neural networks (Deep learning) (Chao, Jun, Yafei, Lingmin, Xiaokang, Jingjing, 2021; Rashe, Zafar, Rasheed, Nouman 2020; Yundong, Cheng, 2016; Mursalin, Eishita, Islam, 2008).

The task of quality control of textile fabrics involves the detection of various types of defects, such as stain, hole, non-stain, creases, edge violation, etc. on a fast-moving material. The automated defect detection system should fix and recognize the defect of the material, its coordinates on the roll, determine the grade of the material according to regulatory documents. The use of a digital high-speed camera to obtain images of the


material gives a large array of input data for an intelligent system. At the same time, it is obvious that the task of machine learning in this case is reduced to the task of classification. This approach makes it possible to determine a discrete class for each defect based on the prepared data.


2 STUDY METHODS


We will analyze the methods of machine and deep learning to determine the most suitable algorithm for solving the production problem of detecting defects in textile fabrics.


Machine learning (ML) is a class of methods whose characteristic feature is not the direct solution of a problem, but learning through the application of solutions to many similar problems. A feature of machine learning algorithms is the use of structured data. That is, the input data array must undergo preliminary processing – filtering, ordering, conversion to a specific format, tabular

^a  <https://orcid.org/0000-0001-7469-041X>

^b  <https://orcid.org/0000-0001-9351-303X>

^c  <https://orcid.org/0000-0003-4029-3509>

^d  <https://orcid.org/0009-0008-2344-2780>

^e  <https://orcid.org/0009-0002-5684-6622>

representation, recovery of missing significant data, etc. Many machine learning algorithms work with numerical data. If categorical features are used, they are translated into numerical format (Brink, Richards, Fetherolf, 2017).

Thus, in the case of using machine learning to detect canvas defects, the system should already have a mechanism for processing and converting data – images coming from the camera. This is advisable only with a limited number of detectable defects.

In artificial neural networks, algorithms work similarly to machine learning, but input data can be interpreted in different ways. That is, data at different levels of the network is determined and identified by different criteria – by analogy with the human brain. That is, the neural network, running data through different levels, eventually finds those identifiers that most adequately determine the input features for the training model. Neural networks show good results in the case of a large volume of unstructured input data.

Let's analyze the array of data obtained in the quality control system of textile materials on disassembling machines. The rewinding speed of materials on typical disassembling machines varies from 15 to 38 m/min (0.25 – 0.63 m/s). The length of the fabric roll can be from 40 to 100 m, knitted fabric 70-90 m. In the case of a digital camera with a shooting speed of 30 frames/s, the microcontroller input will receive from 4800 to 12000 images per roll. For a camera with FPS 12 from 1920 to 4800. In accordance with GOST 161-86, the grade of the fabric is determined by 28 local and 11 common defects. 20 fabric defects are unacceptable and must be cut out.

Thus, in the case of setting the task of detecting the maximum number of defects according to regulatory documents, the number of classes in the classification problem is large, which, with a large number of input frames, leads to the need to use a neural network (Mursalin, Eishita, Islam, 2008).

In addition to data preparation, machine learning includes the stages of model construction – task definition, algorithm selection; model evaluation (cyclic sliding control, performance measurements and parameter adjustment); optimization, scaling and forecasting. Regarding the application for the applied problem of detecting defects in textile fabrics, we will consider a number of classification algorithms based on the technology of teaching with a teacher. In this case, the target variable is available in the training sample (Brink, Richards, Fetherolf, 2017).

The logistic regression algorithm refers to linear algorithms and distributes data into categories based on quantitative features (Brink, Richards, Fetherolf, 2017). Predicts the target class or probabilities of the target classes. Logistic regression is the simplest machine learning algorithm and is well suited for solving the binary classification problem when it is necessary to decompose data into two categories. In this case, the logistic regression is based on a sigmoid, which, regardless of the input value passed to it, always returns a value between 0 and 1:

$$f(x) = \frac{1}{1+e^{-x}}. \quad (1)$$

In general, to build a classifier, you need to find the boundary that best divides the data into target classes (categories). The number of model parameters that need to be determined is determined by the number of measurements in which the resolving boundary is described. On a plane, in two dimensions, it is a line. The separation boundary for the sigmoid is usually assumed to be 0.5.

To assess the accuracy of the model, a statistical deviation is used or an error matrix is constructed to compare truly positive, truly negative, falsely positive and falsely negative indicators in the ML. Gradient descent is used to optimize the model. Despite the fact that the logistic regression algorithm is simple and convenient for calculation, it works well with a large data set, its performance decreases with increasing degree of nonlinearity of the resolving boundary. Another disadvantage of the algorithm is the possibility of retraining the model.

The support vector machine (SVM) method, despite its linear nature, belongs to the class of nonlinear algorithms due to the use of a nonlinear kernel function (V'yugin, 2014). The core is a mathematical construction that can "include" a space with data. In this additional space, the algorithm is able to find a linear boundary, which, after returning to the usual space, turns out to be nonlinear. The algorithm determines a line or hyperplane in the case of a larger number of dimensions that optimally separates the classes of objects. And then calculates the maximum gap between the points on both sides of the solution line.

The main disadvantage of nonlinear algorithms is the need for large computing resources and the problem of scalability to a large amount of data. The probability of generalization error is taken as an estimate. The relation (V'yugin, 2014) can be used as the equation of the optimal hyperplane:

$$(\bar{w}_0 \cdot \bar{x}) - \frac{c_1(\bar{w}_0) + c_2(\bar{w}_0)}{2} = 0, \quad (2)$$

where \bar{w}_0 is the normalizing vector; \bar{x} is the vector of input features; $c_1(\bar{w}_0)$ and $c_2(\bar{w}_0)$ are the parameters of the separating hyperplane:

$$c_1(\bar{w}_0) = \min_{y_i=1}((\bar{w} \cdot \bar{x}_i)), \quad (3)$$

$$c_2(\bar{w}_0) = \max_{y_i=1}((\bar{w} \cdot \bar{x}_i)).$$

The distance from the nearest points of the positive and negative parts of the sample to the optimal hyperplane is the geometric edge of the error, determined by the formula:

$$\rho(\bar{w}_0^*) = \frac{1}{\|\bar{w}_0\|}. \quad (4)$$

The generalization error is determined from the relation linking the number of support vectors d ($d > 2$) and the sample size l :

$$\varepsilon \leq \frac{d \ln l}{l - d}. \quad (5)$$

The k-nearest neighbors method is a nonlinear algorithm with built-in support for multiclass problems (Hastie, Tibshirani, Friedman, 2009). The main idea is to classify new data by comparing them with similar records from the training sample. That is, the analyzed object belongs to the class to which most of its k neighbors from the training sample belong. Usually 3, 5 or 9 neighbors are used and the most common class is selected. Choosing the number of neighbors is important. Too small value of k will lead to the fact that the model will accurately predict on the training set of data, but it will be ineffective for test data. On the other hand, too high a k indicator will make the model too complex.

If the number of classes is greater than 3, then the weighted nearest neighbors method is used, that is, each i-th neighbor is assigned a weight decreasing with the distance to the neighbor. In this case, the object's belonging to the class is determined by the largest total weight among the k nearest neighbors.

When using this algorithm, it is important to normalize the values of features, otherwise one feature with the maximum value may become predominant, and features with small values will not be taken into account in classification.

Euclidean metric, Chebyshev distance, Manhattan distance, Minkowski metric, etc. are used as distance metrics to k neighbors.

In the case of using the Euclidean distance to record a data set consisting of a series of numbers n_i , the distance d between the records can be found by the formula (Brink, Richards, Fetherolf, 2017):

$$d = \sqrt{n_1^2 + n_2^2 + \dots + n_n^2} \quad (6)$$

To get a prediction for new records, we look for the nearest known record and assume that the new record belongs to the same class. This is a classifier based on one nearest neighbor, since only one nearest record is considered.

The advantage of the k-nearest neighbors method is to work with any number of categories in the classification problem, as well as good scalability. At the same time, the model has only two parameters: the number of neighbors k and the distance metric. The disadvantages of the method include large computational costs, as well as the complexity of working with categorical data.

The k-nearest neighbor method works well when training needs to be done quickly, and the prediction itself by the model is slower.

Decision trees are a highly accurate nonlinear algorithm widely used to solve classification problems. The algorithm allows the computer to determine from the training sample which variables are the most important. It is these that it places at the top of the tree, then gradually moving on to less important variables (V'yugin, 2014). It has a number of other advantages, for example, insensitivity to insignificant signs, noisy datasets with missing values.

The Boosting algorithm is used in machine learning to reduce the number of errors in predictive data analysis. It builds ensembles of models by sequentially combining several decision trees. Weights are assigned to the output data of individual trees. Then incorrect classifications from the first decision tree are assigned more weight, after which the data is transferred to the next tree. After numerous cycles, boosting combines weak classifiers into one powerful prediction algorithm.

A simple Naive Bayesian classification algorithm is used in text recognition tasks. It is based on Bayes' theorem and assumes the independence of features from each other. The advantage of the naive Bayesian classifier is the small amount of data required for training, parameter estimation and classification.

Analyzing the above-mentioned machine learning algorithms in relation to solving the problem of detecting defects on moving textile fabrics, we come to the conclusion that they are

applicable only in the case of a limited number of defined defects in a narrow range of their variations. For a full-fledged intelligent quality control system of the material, determining its grade, it is necessary to use an algorithm that would solve the classification problem taking into account the fuzzy structuring of the input data. This algorithm is implemented by an artificial neural network

The neural network algorithm is used to evaluate unknown functions based on a large amount of input data through back propagation. It surpasses the other methods in terms of complexity, but in certain cases it gives excellent results. It is the basis of many deep learning methods (Nikolenko, Kadurin, Arhangel'skaya, 2018; Krejman, 2022; Tarik, 2019).

In general, a neural network is a computer program that has certain characteristics and a specific algorithm construction. Unlike a regular program with a linear algorithm, it works like a human brain. To solve image processing problems, including in the field of quality control of textile products, a special architecture is used – a convolutional neural network, which was proposed by Jan Lekun and designed for problems of classification of graphic images. Such a network scales well enough and can be used for pattern recognition of any resolution (Nikolenko, Kadurin, Arhangel'skaya, 2018).

To recognize an image, the neural network must be trained on some sets of input data or objects, each of which belongs to one of k numbered classes (0,1,2, ... $k-1$). Therefore, after training, the resulting model should classify images unknown to it from test sets, that is, indicate which class this object belongs to (Tarik, 2019).

3 RESEARCH RESULTS

In this paper, a solution is proposed for the implementation of a neural network in a textile materials quality assessment system implemented on a Raspberry Pi microcontroller with a Raspberry Pi Camera V1.3 connected to it. The software (neural network) is implemented in the Python programming language in the Google Colab cloud development environment, using TensorFlow and Keras libraries.

Images from the camera are transmitted to the neural network of the microcontroller, they are analyzed and defects are determined. A knowledge base of defects was developed in accordance with GOST 161-86, which will allow determining the

grade of a piece according to all the defects listed in the normative document: stain, hole, non-stain, pile bald spots, creases, marking-off, wrinkle, miss-fitting, bowing, edge violation, etc. (Kaznacheeva, Stupak, 2022).

Neural networks require a large amount and quality of data. There are ready-made solutions, but for the task of recognizing defects on the fabric, the data set was created manually. It contains a tree of folders, each of which stores images with defects. For clarity, defects such as a spot and a hole were selected.

One of the options for creating a neural network in textile quality control tasks is its implementation in the Matlab software environment using the Deep Learning Toolbox module, which supports the development of applications for machine learning, as well as applications for convolutional neural networks (Amos, 2016). To solve this problem, an example of a simple classification and deep learning network was chosen. As a result of the conducted research, the accuracy with which the created neural network works was 60%. The accuracy of the model used is not high enough, so it requires a procedure to improve and optimize it.

Another tool for creating a neural network is the high-level programming language Python. The advantages of Python include simplicity and versatility, the ability to load program code into the Raspberry Pi microcontroller, as well as the availability of libraries suitable for the development of the desired model (Hastie, Tibshirani, Friedman, 2009; Kaznacheeva, Stupak, 2021; Kaznacheeva, Zaharkina, Vlasenko, Ryzhkova, 2021). Such libraries are TensorFlow and Keras (Klette, 2019; Solem, 2016). They are characterized by open source code and perform numerical calculations using data flowcharts. Provide a simple and convenient way to create deep learning models.

The neural network was developed in the interactive cloud environment of Google Colab. The service is based on the Jupyter Notebook toolkit for working in Python. At the first stage of development, it is necessary to install library modules, and upload a pre-prepared archive with data to the project. The data set contains 2 classes of defects: a spot and a hole (Tarik, 2019; Kaznacheeva, Zaharkina, Vlasenko, Ryzhkova, 2021). There are 2000 images of defects in total.

To load a set of images into Tensorflow, the `image_dataset_from_directory` utility is used. It creates a `tf.data.Dataset` file from a directory with images, which has a specific structure, where

images of the same class are located in a separate directory (Kaznacheeva, Stupak, 2022).

At the second stage, a directory is formed in which a new data set is created. The catalog is divided into two parts:

- training dataset (1800 images);
- testing dataset (200 images).

When using the `image_dataset_from_directory` utility, the `validation_split` parameter is entered, which is responsible for forming a subset responsible for checking the operation of the neural network. The number of images included in the subset is 10% of the total amount of data used for training. The size of each image is 277×277 pixels. Next, a list with class names is formed, and several

images from the "Defects" dataset are displayed (Fig. 1).

At the third stage, a neural network for recognizing defects is determined. Fig. 2 shows the structure of the neural network. In its standard architecture, convolutional layers and subsampling layers alternate. At the output, 2 neurons are formed according to the number of classes in the data set. Figure 3 shows information about the learning outcomes: the number of epochs is 5; the learning time of one epoch is approximately 50 seconds; the maximum percentage of correct answers is 72%.

To assess the quality of the model on the test data set, the `model.evaluate` method is used (Fig. 4). As can be seen from the figure, the proportion of correct answers to the test data was 75%.

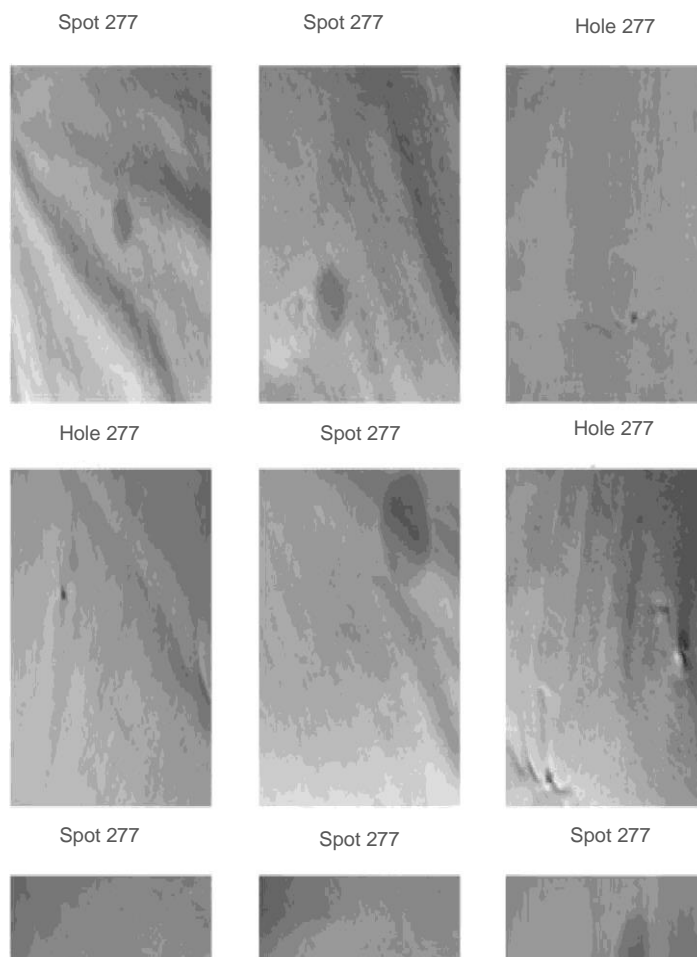


Figure 1: Output of defect images.

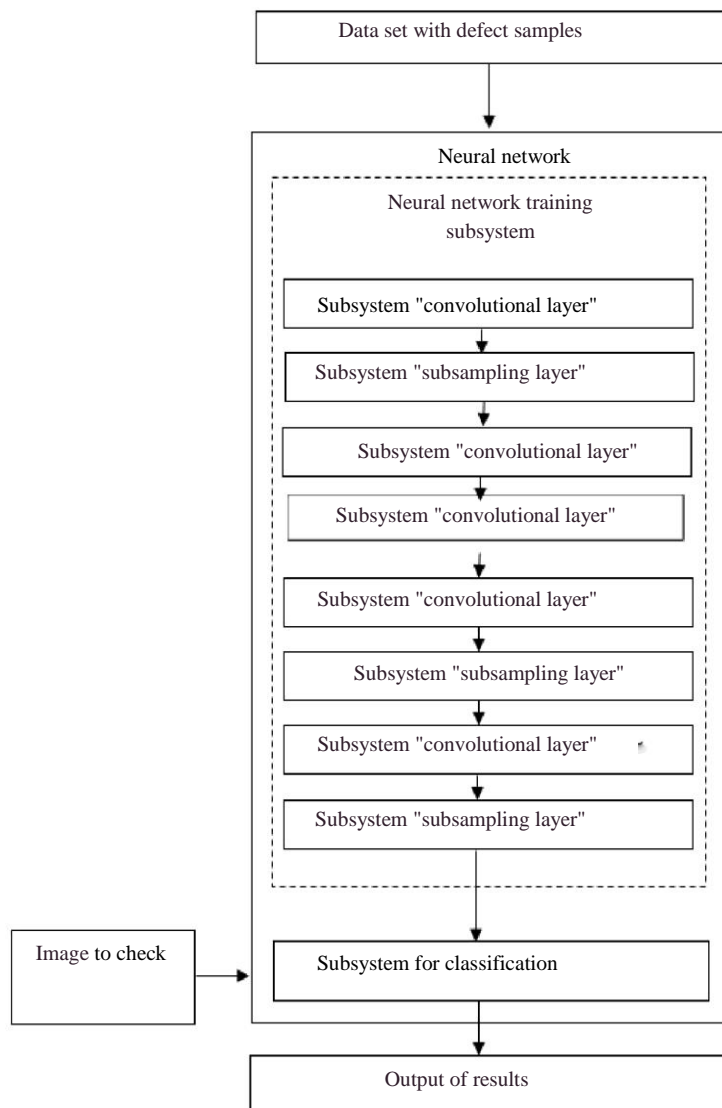


Figure 2: Neural network architecture.

Epoch 1/5
 33/33 - 50s - loss: 34.3603 - accuracy: 0.4691 - val_loss: 0.6873 - val_accuracy: 0.5000 - 50s/epoch - 2s/step
 Epoch 2/5
 33/33 - 49s - loss: 0.6934 - accuracy: 0.5864 - val_loss: 0.6950 - val_accuracy: 0.5000 - 49s/epoch - 1s/step
 Epoch 3/5
 33/33 - 48s - loss: 0.6962 - accuracy: 0.5679 - val_loss: 0.6832 - val_accuracy: 0.6667 - 48s/epoch - 1s/step
 Epoch 4/5
 33/33 - 48s - loss: 0.6945 - accuracy: 0.5432 - val_loss: 0.6806 - val_accuracy: 0.7222 - 48s/epoch - 1s/step
 Epoch 5/5
 33/33 - 48s - loss: 0.6886 - accuracy: 0.5802 - val_loss: 0.6742 - val_accuracy: 0.5556 - 48s/epoch - 1s/step

Figure 3: Neural network training.

```
[57] # We evaluate the quality of model training based on
test_data
scores = model.evaluate(test_dataset, verbose=1)

3/3 [=====] - 1s 310ms/step - loss: 0.5751 - accuracy: 0.7500

[58] print("The percentage of correct answers to test data, as a
percentage:", round(scores[1] * 100, 4))

The percentage of correct answers to the test data, as
a percentage: 75.0
```

Figure 4: Evaluation of the quality of neural network training.

Figure 5 (a, b) shows graphs of the quality of neural network training

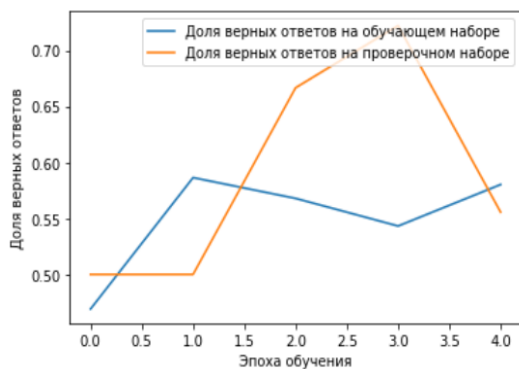


Figure 5 (a): Graphs of correct answers on the training set.

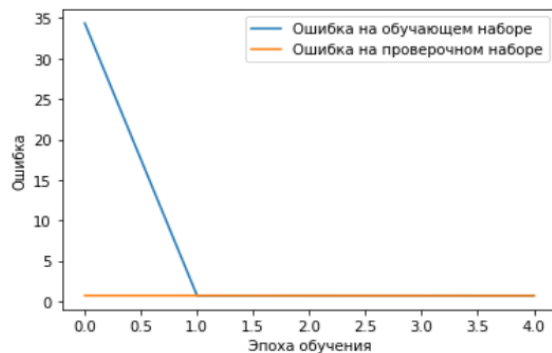


Figure 5 (b): Error graphs on the training set.

The result of checking the operation of the obtained model is the recognition of the defect image by classification criteria. To do this, you need:

- to download the program to your computer as a Hierarchical Data Format (HDF) file;
- to add this image to the model.

After loading the image, the Image command is executed (Fig. 6). The image dimensions in pixels are specified as parameters.

```
[74] Image(img_path, width=277, height=277)
```



Figure 6: Output of the downloaded defect image to the screen.

Figure 7 shows the result of the trained neural network. At the output, two neurons are formed by the number of classes. That is, of the two available classes, the image most likely belongs to the class with the number 0 and the name "Hole 277". The np.argmax method is used to print the class number and name.

```

prediction =
np.argmax(prediction)
print("HoMep class:",
prediction) print("Ha3BaHMe
class:", classes[prediction])

Class Number: 0
Class name: Hole
277

```

Figure 7: Displaying the class number and name on the screen.

5 RESULT DISCUSSION

In the case of using the Deep Learning Toolbox module and applications for convolutional neural networks of the Matlab software environment for the implementation of a neural network for the task of quality control of textiles, the use of the classification algorithm gave the accuracy of the neural network 60%. The model used requires a procedure for its improvement and optimization.

The Python development environment with built-in TensorFlow and Keras libraries is a more powerful tool for machine learning and neural network development. An important advantage of this tool is the ability to download program code into the Raspberry Pi microcontroller.

The article describes the process of developing, training and testing a neural network, which allowed us to obtain the following results: the number of epochs is 5; the training time of one epoch is approximately 50 seconds; the maximum percentage of correct answers is 72%.

The result of checking the operation of the obtained model is the recognition of the defect image according to classification criteria with the proportion of correct answers to the test data of 75%.

6 CONCLUSIONS

Thus, in relation to solving the problem of detecting various defects on moving textile fabrics, neural networks are the most powerful and effective means, and the rational software for creating a neural network is the Python development environment with built-in libraries for deep machine learning TensorFlow and Keras. This programming environment is cross-platform, capable of working with multiple hardware platforms or operating systems, and also allows you to load program code directly into the microcontroller.

REFERENCES

- Chao, L., Jun, L., Yafei, L., Lingmin, H., Xiaokang, F., Jingjing, Ch., 2021. Fabric Defect Detection in Textile Manufacturing: A Survey of the State of the Art. Security and Communication Networks. Volume 2021, Article ID 9948808, 13 pages.
- Rashe, A., Zafar, B., Rasheed, A., Nouman A., 2020. Fabric Defect Detection Using Computer Vision Techniques: A Comprehensive Review. Mathematical Problems in Engineering Volume 2020, Article ID 8189403, 24 pages.
- Yundong, L., Cheng, Zh., 2016. Automated vision system for fabric defect inspection using Gabor filters and PCNN. Li and Zhang SpringerPlus, 5:765 pages.
- Mursalin, T. E., Eishita, F.Z., Islam, A.R., 2008. Fabric defect inspection system using neural network and microcontroller. *Journal of Theoretical and Applied Information Technology*. pages 560 – 570.
- Brink, H., Richards, J.W., Fetherolf, M., 2017. Real -World Machine Learning. Shelter Island, NY: Manning Publications Co. 298 pages.
- V'yugin, V.V., 2014. Matematcheskie osnovy mashinogo obucheniya i prognozirovaniya. 304 pages.
- Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning. Data Mining, Inference, and Prediction, Second Edition. Springer Science, 737 pages.
- Nikolenko, S., Kadurin, A., Arhangel'skaya, E., 2018. Glubokoe obuchenie. 480 s.
- Krejman, G., 2022. Biologicheskoe i komp'yuternoe zrenie. pages 314.
- Tarik, R., 2019. Sozdaem nejronnyuy set'. pages 272.
- Kaznacheeva, A.A., Stupak, M.M., 2022. Razrabotka bazy znaniy specialista predmetnoj oblasti kontrolya kachestva tekstil'nyh materialov. Dizajn, tekhnologii i innovacii v tekstil'noj i legkoj promyshlennosti (INNOVACII-2022): sbornik materialov Mezhdunarodnoj nauchno-tehnicheskoy konferencii. CHast' 2. pages 249-253.
- Amos, G., 2016. MATLAB. Teoriya i praktika. DMK-Press: M. pages 416.
- Kaznacheeva, A.A., Stupak, M.M., 2021. Obzor gotovyh bibliotek realizacii nejronnoj seti v algoritmah avtomatizirovannogo obnaruzheniya defektov na tkani. Sbornik nauchnyh trudov kafedry avtomatiki i promyshlennoj elektroniki Rossijskogo gosudarstvennogo universiteta im.
- Klette, R., 2019. Komp'yuternoe zrenie. Teoriya i algoritmy. DMK-Press: M. pages 506.
- Solem, YA., 2016. Programmirovaniye komp'yuternogo zreniya na yazyke Pyton. DMK-Press: M. pages 311.
- Kaznacheeva, A.A., Zaharkina, S.V., Vlasenko, O.M., Ryzhkova, E.A., 2021. Razrabotka avtomatizirovannoj sistemy obnaruzheniya defektov na tkani s primeneniem komp'yuternogo zreniya. *ZHurnal «Inzhenernyj vestnik Dona» №12*. pages 234-245.