Research and application of neural network approaches to solving image recognition problems

Marina Bolsunovskaya1, Egor Kuzmichev1, Svetlana Shirokova1*, and Tojiboy Almataev2

1Peter the Great St. Petersburg Polytechnic University, 29, Polytechnicheskaya, Saint-Petersburg, 195251, Russian Federation
2Andijan Machine-Building Institute, 56, Bobur Shoh St., Andijan, 170019, Uzbekistan

Abstract. The paper investigates neural network approaches to solving number recognition problems and develops an algorithm for creating authentic datasets. In the course of the work, research and development of an algorithm for creating authentic datasets for solving the problem of number recognition is carried out. When solving image recognition problems, it is advisable to use neural network technologies, but often there is a problem of lack of data to form a full-fledged training sample during recognition. An algorithm has been developed to create a set of artificial data appropriate for use in training neural networks. The recognition of number plates and wagon numbers is assumed to be the scope of application. An algorithm that forms a set of synthetic images marked up for training has been created. The result of the algorithm application is a dataset appropriate for supplementing the training sample when training neural networks in the field of recognition of number plates and wagon numbers.

1 Introduction

Artificial neural networks have been increasingly used in recent years. The use of this technology makes it possible to solve previously insurmountable or unnecessarily costly tasks, however, a problem of lack of data for the formation of a full-fledged training sample often arises in the course of work. This issue is especially relevant in the field of object recognition, since additional training materials are often extremely difficult to get.

The way to solve the problem of filling the training data shortage is the use of synthetic data, however, it is necessary to take into account the need to generate authentic data, since the result of using not fully correct input data is a defective neural network.

To achieve this goal, it is necessary to develop an algorithm capable of forming a dataset appropriate for training a neural network. Since real data is also used when training a neural network, an artificial dataset must comply with the rules of marking up of the original one, expand recognition capabilities without having a significant negative impact on recognition accuracy [1,2].

* Corresponding author: swchirokov@mail.ru
Currently, neural networks are becoming more widespread in many areas, as they allow solving complex problems and improving existing solutions. Of particular importance in the process of working with neural networks is the stage of their training.

The process of preparing a dataset is an important stage, as a result of which a processed set of cleaned data is obtained, appropriate for processing by machine learning algorithms. A dataset for training a neural network is processed and structured information in tabular form [3].

The use of an appropriate dataset is of particular importance, since the correctness of the neural network depends on the results of training. Nevertheless, the problem of lack of existing data often arises on the way to forming the most complete and accurate dataset. This problem is particularly noticeable in the field of pattern recognition, since obtaining new real data can be difficult or impracticable.

Thus, there is a problem of filling gaps in training, the solution of which is the use of synthetic data. Speaking about the synthesis of a training sample, it is necessary to take into account a number of features, both theoretical and practical [4,5]. On the one hand, synthetic data should strive to match the real ones as completely as possible. On the other hand, creating a dataset is a costly process both in terms of using computer resources and in terms of time. Therefore, artificial data is not required to fully correspond to real data, but all the signs that the neural network pays attention to should be as close to the real ones as possible.

Since the recognition of wagon numbers acts as a subject area, this requirement can be specified as follows: synthetic data should not have a photographic resemblance to real images, but must be able to reliably simulate all possible features of their appearance.

Due to the fact that the purpose of creating and using synthetic data in training is to expand the capabilities of a neural network, fluctuation in the accuracy of training in a particular case is an expected outcome, but the introduction of an artificial dataset should not have a significant negative impact. In other words, when testing on the same validation dataset, a neural network trained with partially synthetic data should not be significantly inferior to a neural network trained on completely real data. Deterioration of recognition accuracy is possible, for example, in the case when the test and training datasets are created using extremely similar images, but even in this case, the use of partially synthetic datasets should not interfere with the neural network to effectively cope with its task.

Also, since the algorithm being developed is supposed to be used in conjunction with neural networks, it is recommended to use languages and services used in working with recognition systems in development. Such an approach will increase the ease of understanding of the generation program by a user qualified enough to work with neural networks.

Thus, the general requirements can be summarized as follows:
1) Artificial data should display all the characteristic features of real data;
2) The use of synthetic data should not impair the recognition results in a particular case;
3) The algorithm is recommended to be implemented via the tools used in working with neural networks;

The next stage is the analysis of development environments and setting specific requirements for the software being created [6,7].

2 Materials and methods

When recognizing number plates, neural networks of two levels are often used – object detection and number recognition.
Wagon numbers have plenty of features that must be taken into account when forming an artificial image:

- A number consisting of eight digits;
- Dirt and scuff degree;
- Number font;
- Breaks and tilts allowed when writing the number;
- Low resolution of the camera used for number recognition;

The system being developed should be able to implement these features.

Since YOLOv5 is used as the basic neural network, the synthetic dataset must be compatible with the real one.

A significant part of neural networks, including the basic YOLOv5 neural network, is written in the Python programming language. This programming language is easy to learn, and also has an extensive set of tools for working with neural networks. Since the program being developed is designed to work with neural networks, Python was chosen as the development language.

The Supervised service was chosen as a service for training neural networks and storing datasets. This platform is free and has a set of tools for creating Python programs.

The Supervised service has a number of features [8]:

1) Powerful annotation sets.

Supervised, equipped with advanced annotation tools, provides a complete set of features that distinguish it from other labelling tools.

2) Collaboration with data and users.

In addition to the usual tools such as rectangle or brush, Supervised comes with “smart” labelling tools based on a set of class-independent neural networks that can be additionally trained on data.


Supervised is probably the only machine learning platform that not only provides easy and convenient access to modern machine learning models and tools directly in the browser, but also combines hundreds of previously isolated projects on one platform in the form of Supervised applications.

The service aims to cover every single use case and aspect of the machine learning pipeline, not just a few of the most common ones, creating a real platform that works as an OS for computer vision.

4) Platform and setup.

Unlike other companies that use monolithic black boxes with virtually no configuration and community involvement, Supervisely provides a real platform that gives the basis for developing and running applications for everyone — just like in an OS such as Windows or macOS.

5) Corporate-level solutions.

The Supervisely Enterprise Edition (EE) is designed for companies that want to scale their artificial intelligence infrastructure, available both in the cloud and in a self-installation.

Sparse R-CNN is a purely sparse method for detecting objects in images. Existing object detection works depend heavily on dense object candidates, such as k reference blocks, predefined on all map grids of image objects of size H× W.

Within this method, a fixed sparse set of sentences of studied objects with a total length of N is provided to the object recognition head to perform classification and location determination. By eliminating HWk (up to hundreds of thousands) candidates for manually created objects, up to N (for example, 100) trainable sentences, sparse R-CNN completely avoids all costs associated with designing candidate objects and assigning labels "many to
one”. More importantly, the final predictions are output directly without postprocedure with maximum suppression.

Since an artificial set of training data should strive for a high degree of authenticity, there are several options for the development of the algorithm:

1) The formation of synthetic data based on real input.

This approach involves using the provided dataset to form a new one based on it. It is assumed that the created data will be as close as possible to the original ones, so that the algorithm will expand the training dataset, but there is a possibility of severe restrictions when generating data.

2) Creating fully synthetic data.

The formation of a dataset without source data is a more difficult task from a technical perspective, but presumably allows getting almost unlimited possibilities in generating a dataset. However, the created data will differ significantly from the real ones, as a result of which there is a risk of obtaining data of unacceptable quality [9].

3 Results

Initially, an algorithm was developed using the images included in the original dataset.

The figures of the real marked-up image were rearranged in random order and sent to the service to be added to new training samples.

Despite the fact that the prototype algorithm allows solving the problem of expanding training datasets and has certain advantages like the expected authenticity of numbers, the use of this approach has revealed a number of characteristic disadvantages:

1) Limiting the digital set of generated images to those presented in the original image;

2) The presence of graphic defects in the original images, limiting the variety of new ones;

3) A limited number of colors and fonts;

The formulated shortcomings determined the need to abandon the use of this algorithm and switch to creating images without using the source data.

Since the result of the algorithm for generating datasets based on existing images does not have sufficient breadth of possibilities, a full generation algorithm has been developed. On the one hand, fully synthetic figures differ somewhat from real ones, but, on the other hand, the use of various processing methods makes it possible to almost completely eliminate this disadvantage, as well as significantly expand the coverage of cases used in neural network training [10].

Interaction with the algorithm is carried out via the graphical user interface (Figure 1):
The interface provides access to the following features:

1) Launch of image generation with parallel upload to the Supervised service using a given key;
2) preview of an example image created based on the specified parameters;
3) creating a folder name for the service Supervisely;
4) specifying the size of the dataset to be created;
5) editing the maximum possible value of the angle of inclination of the digits;
6) image blur editing;
   - the value of the maximum possible blurriness of the image;
   - the value of the probability of applying blur during generation;
7) editing the probability of creating faintly distinguishable images;
8) editing the use of the mud effect on the number plate;
   - the maximum number of foci of the effect spread;
   - the probability of using the effect;
   - extent of distribution;
9) editing the use of the effect of scuffing numbers on the number plate;
   - the maximum number of foci of the effect spread;
   - the probability of using the effect;
   - extent of distribution;
10) the image noise level;
11) editing gaps between digits;
   - maximum number of breaks;
   - the probability of a gap.

The current algorithm first generates a random number, after which it creates an image marked up for training with the necessary effects [11]. The resulting dataset can be used for training neural networks.

When forming an image based on the generated number, fonts applying numbers to existing trains are used. Each digit is sequentially applied to the background of a randomly selected color, when applied, small fluctuations along the Ox and Oy axes that occur when numbering cars are simulated. The result is an image appropriate for subsequent processing.

After creating the base image, the algorithm applies additional effects if they were requested by the user when launching the image and the probability of application allows them to be used.
Due to the fact that the camera that registered the train with the recognized number, or the image itself may have a low resolution, the program allows adding a blur effect to the images.

Upon applying all the required effects, if the user has selected the dataset generation mode, the image is marked up for use in training. Markup data is collected throughout the image creation process [12].

In the course of the work, two algorithms were implemented, the results of which turned out to be similar to the expected ones.

As expected, the program for creating datasets based on available images has significant limitations that do not allow it to be used fully to achieve its objectives. Nevertheless, the achievements of this algorithm can be applied in a limited number.

The program for generating synthetic datasets allowed us to fully achieve our goals, showing ample application opportunities.

The algorithm verification is carried out using the YOLOv5, TOOD and R-CNN neural networks.

Since the YOLOv5 neural network is fundamental, its testing is carried out in more detail, the TOOD and R-CNN networks are used for additional verification of the result.

Testing of the YOLOv5 neural network is carried out in two stages:
1) Testing on a neural network trained without prior validation.
2) Testing on a neural network trained using additional validation [13].

After receiving the recognition accuracy results, the number of recognized digits is checked using an additional validation dataset. This set is common to all learning outcomes and does not contain synthetic images and images used in training. The TOOD and R-CNN networks were tested only for the training quality assurance.

Also, there was a fluctuation in the number of directly recognized digits, and in most cases the indicators of synthetic datasets were not inferior to “pure” datasets, and sometimes exceeded them. Since the purpose of creating synthetic data sets is to expand the recognition capabilities of a neural network, and a theoretical decrease in recognition accuracy in a particular case is an expected and acceptable result, we can state the success of verifying the results of using the algorithm.

### 4 Discussion

Since the purpose of introducing synthetic images is to expand the recognition capabilities of the neural network, it is assumed that the recognition accuracy fluctuation does not exceed 5%.

The results of testing using the YOLOv5, TOAD and CNN neural networks showed an insignificant fluctuation in the accuracy of digit recognition within 5%, which is an acceptable error.

### 5 Conclusions

In the course of the work, the problem of the lack of training data for training a neural network was considered and the task was set to fill in the missing data through the formation of synthetic datasets. Recognition of train numbers was chosen as the field of application.

The general requirements for solving the task were identified, and the relevance of work in the field under study was emphasized.

In the process of domain analysis, the requirements were specified, Python was chosen as the development language because of its close integration with neural networks.
In the course of solving the problem, two algorithms were implemented – an algorithm for generating datasets based on existing images and an algorithm for generating synthetic datasets. The results of the first program had significant drawbacks, so the program for generating synthetic datasets was further developed.

The results of the program for generating synthetic data sets were tested on the example of the YOLOv5, TOOD and R-CNN neural networks. During testing, an insignificant fluctuation in the accuracy of digit recognition within 5% was detected, while the quantity of recognized numbers remained at the same level or increased, and trained neural networks became more versatile due to training on new input data, from which it can be concluded that the training was successful and the goal was achieved.

Acknowledgment

The research is funded by the Ministry of Science and Higher Education of the Russian Federation under the strategic academic leadership program “Priority 2030” (Agreement 075-15-2021-1333 dated 30.09.2021).

References


11. O. G. Kantor, S. I. Spivak, Informatics and its Application (Informatika i ee Primenenie) 8(2), 111-121 (2014)


13. V. N. Volkova, Theory of information systems. Textbook (Teoriya informacionnykh sistem. Uchebnoe posobie) (Saint-Petersburg Polytechnic University Publ., Saint-Petersburg, 2014)