ИНСТИТУТ СИСТЕМНОГО АНАЛИЗА И УПРАВЛЕНИЯ

COLLECTION OF SCIENTIFIC AND PROJECT ACTIVITY REPORTS OF THE GRADUATES OF THE INTERNATIONAL SCHOOL OF INFORMATION TECHNOLOGIES "DATA SCIENCE"

DATA SCIENCE

IT SCHOOL

Ministry of Education of the Moscow Region

Dubna State University

Joint Institute for Nuclear Research

COLLECTION OF SCIENTIFIC AND PROJECT ACTIVITY REPORTS OF THE GRADUATES OF THE INTERNATIONAL SCHOOL OF INFORMATION TECHNOLOGIES "DATA SCIENCE"

VOLUME 2

Edited by V. Korenkov, Eu. Cheremisina, O. Streltsova, D. Priakhina, A. Stadnik

DATA SCIENCE

IT SCHOOL

DUBNA

2021

УДК 004.42
ББК 32.97я43
С70

Translators: Julia Meshcherskaya, Ekaterina Voytishina

С70 **Collection of scientific and project activity reports of the graduates of the International School of Information Technologies “Data Science”**. Volume 2 / edited by V. Korenkov et al. – Dubna : JINR, 2021. – 48 p.

The publication presents a collection of brief summary reports of the State Budgetary Educational Institution of Higher Professional Education of the Moscow Region “Dubna University” students who studied at the International School of Information Technologies “Data Science” from 2019 to 2021.

During the studies, the students were involved in real promising projects of the Joint Institute for Nuclear Research (JINR, Dubna, Russia) and other organisations, the work in which was performed by the students within the discipline “Scientific projects” under the supervision of the JINR and Dubna University staff.

УДК 004.42
ББК 32.97я43

CONTENTS

PREFACE

The International School of Information Technologies "Data Science" (hereinafter referred to as the IT School) is a joint educational project of the Meshcheryakov Laboratory of Information Technologies (MLIT) of the Joint Institute for Nuclear Research (JINR) and the Institute of System Analysis and Management (ISAM) of Dubna State University. Its goal is to train highly qualified IT specialists for the development of megaprojects' computing, Big Data analytics, the digital economy and other promising areas.

The educational program of the IT School is formed taking into account the personnel needs of JINR and other organizations of the high-technology sector of the economy and is implemented with their participation. The program comprises such disciplines as:

- Additional chapters of mathematics;
- Programming languages for data analysis;
- Introduction to Unix operating systems;
- Tools for the collaborative development of software;
- Introduction to cloud technologies;
- Big Data analytics;
- Applied tasks of data analysis;
- Distributed systems;
- Multi-agent systems;
- High-performance computing;
- Modern data processing and storage methodologies;
- Quantum software engineering;
- Professional English.

To conduct classes with students at Dubna University and MLIT JINR, an educational infrastructure was created. It consists of two computer rooms, a conference hall and a room for individual work with students. Practical classes are held in computer rooms, involving the resources of the HybriLIT heterogeneous computing platform, which is part of the Multifunctional Information and Computing Complex of MLIT JINR.

Staff members of Dubna University and JINR participate in the organization of the educational process and in the creation of a software and information environment.

The second enrollment of students to the IT School took place in September 2019. Students of Dubna University who wanted to enter the IT School passed a competitive selection, which covered three stages, namely, a questionnaire, a written exam and an interview. Training at the IT School is free of charge. The educational program of the IT School is mastered by students in parallel with the main educational program.

The IT School closely collaborates with leading Russian universities that train qualified IT specialists. Therefore, during the studies, students gained knowledge and competences in the field of modern computing and Big Data analytics not only from teachers of Dubna University and the JINR staff. Lectures were also given by teachers from such Russian universities as the National Research Nuclear University MEPhI, Plekhanov Russian University of Economics, etc.

Moreover, students attended lectures and workshops from leading experts of Russian companies:

- workshop on the topic "Intel architectures and technologies for high-performance computing and the tasks of machine/deep learning (ML/DL)" by specialists of the Intel and RSC companies, on the basis of MLIT JINR (15 November 2019);
- tutorial on the topic "Working with convolutional neural networks in the Keras framework" by a representative of the "Videointellect" company (28 November 2019);
- workshop on high-performance computing on the basis of the National Research University "Higher School of Economics" (21 January 2020);
- lecture on the topic "Trends and principles of computing. Modern basic components of building computer systems" by the director of development of HPC/Cloud corporate projects of the Intel company) (27 November 2020).

Students of the IT School took an active part in educational and scientific events for students:

- School of Young Scientists "High-performance platforms for the digital economy and mega science projects" (3-4 December 2019, PRUE, Moscow, Russia);
- II School of Young Scientists "High-performance platforms for the digital economy and mega science projects" (17-18 November 2020, PRUE, Moscow, Russia);
- XXVIII Scientific-practical Conference of students, postgraduate students and young specialists (12-23 April 2021, Dubna University, Dubna, Russia).

Due to the current epidemiological situation since the beginning of 2020 and the transfer to remote learning, students participated in some events via videoconferencing.

The second graduation of students of the IT School took place in June 2021. Ten graduates successfully completed their studies and were awarded certificates of additional education within the program "Data Science".

One of the major principles of the IT School rests on training through research. Therefore, during the studies, students were involved in JINR's real promising projects, the work in which was performed within the discipline "Scientific projects". The given publication presents a collection of brief reports on their activities.

The Directorate of the IT School expresses its gratitude to the teachers of Dubna University and the JINR staff for their fruitful work with students.

The following specialists worked with students of the IT School:

- Balashov N., software engineer of MLIT JINR;
- Belov M.A., associate professor of the department of SAM ISAM of Dubna University;

- Gertsenberger K., candidate of technical sciences, scientific and experimental department of heavy-ion collision physics at the NICA complex, head of the group of mathematical support and software of VBLHEP JINR;
- Ivantsova O., senior teacher of the department of SAM ISAM of Dubna University;
- Kadochnikov I., software engineer of MLIT JINR;
- Kalinovsky Yu., doctor of physics and mathematics, leading researcher of MLIT JINR;
- Koshlan D., software engineer of MLIT JINR;
- Meshcherskaya J., candidate of pedagogical sciences, associate professor of the department of SAM ISAM of Dubna University;
- Ososkov G., doctor of physics and mathematics, principal researcher of MLIT JINR;
- Pelevanyuk I., researcher of MLIT JINR;
- Reshetnikov A., candidate of technical sciences, associate professor of the department of SAM ISAM of Dubna University;
- Stadnik A., candidate of physics and mathematics, senior researcher of MLIT JINR;
- Sychev P., associate professor of the department of DICS ISAM of Dubna University;
- Tyatushkina O., candidate of technical sciences, associate professor of the department of SAM ISAM of Dubna University;
- Ulyanov S., doctor of physics and mathematics, professor of the department of SAM ISAM of Dubna University;
- Zrelov P., candidate of physics and mathematics, head of the scientific and technical department of software and information support of MLIT JINR.

Scientific leaders of the IT School:
V. Korenkov*, doctor of technical sciences, director of MLIT JINR,*
head of the department of DICS ISAM;
Eu. Cheremisina*, doctor of technical sciences, professor, academician of RANS,*
director of ISAM.
Director of the IT School:
O. Streltsova*, candidate of physics and mathematics, senior researcher of MLIT JINR,*
associate professor of the department of DICS ISAM.
Scientific secretary of the IT School:
D. Priakhina*, researcher of MLIT JINR,*
senior teacher of the department of DICS ISAM.

itschool.jinr.ru

GRADUATES OF 2021

1. Bachinsky Stanislav
2. Kolobov Sergey
3. Melnik Nikolay
4. Nechushkin Kirill
5. Papoyan Georgy
6. Rybakina Alena
7. Sasin Alexey
8. Sokolova Maria
9. Khodyrev Ivan
10. Tsegelnik Nikita

DATA MINING

DEVELOPMENT OF AN AGENT SYSTEM FOR ASSESSING THE PUBLICATION ACTIVITY OF THE BRICS COUNTRIES

Sokolova Maria[1], Koshlan Diana[2], Priakhina Daria[3]

[1] *Student;*
Dubna State University;
The International School of Information Technologies
"Data Science";
The main educational program:
Software Engineering, group 4251;
e-mail: somv.17@uni-dubna.ru.

[2] *Software Engineer;*
Meshcheryakov Laboratory of Information Technology;
Joint Institute for Nuclear Research.
PhD student;
Dubna State University.

[3] *Researcher;*
Meshcheryakov Laboratory of Information Technology;
Joint Institute for Nuclear Research.
Senior Teacher;
Department of Distributed Computer Systems;
Dubna State University.

Keywords: agent program, information resources, publication activity, BRICS union.

Introduction

In each of the member countries of the BRICS union (Brazil, Russia, India, China, South Africa), scientists are engaged in scientific work in different directions, the results of which are mainly reflected in the articles. In order to analyze the contribution of each country of the union to the world science, to identify the leading scientific directions of the states, to evaluate the publication activity of various organizations and universities, to identify possible areas of cooperation between organizations of different countries, it is necessary to develop a system that allows collecting information from databases that contain information about scientific publications published in leading journals, and to provide statistical information on the scientific and technological development of the BRICS countries in a convenient form for analysis.

The purpose of the work is to collect statistical data of publications of each country of the BRICS Union and present this data in the form of a web application for their subsequent analysis.

1. Data collecting

The data source is the Nature Index website [1], which aggregates links to publications in various journals for each country. To collect data, an agent program has been created that automatically scans predefined sites on the Internet, collects information based on search instructions and delivers it to the appropriate section of the database.

To implement the agent program, the Python programming language and the Selenium WebDriver browser management library are selected.

The tasks of the agent program include collecting publication data for each country of the BRICS Union using the Nature Index information resource and delivering the collected data to the storage system. The algorithm of the agent is shown in Figure 1.

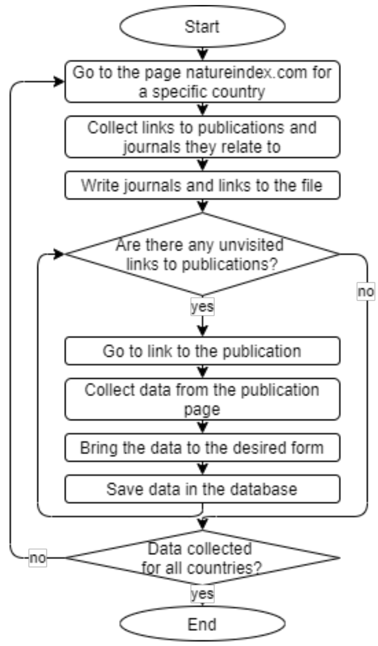


Fig. 1. The algorithm of the software agent

To collect publication data, the agent must go to the publication page by clicking on the link for it. The information resource natureindex.com contains a separate page for each country, where a list of topics is presented. Each topic is expanded into a list of journals that publish scientific articles. Each journal from the list is expanded into a list of links to publications, in turn, the links lead to a page of the same information resource natureindex.com with the summary of the publication and a button to go to the publication page on the website of a particular journal.

To make it possible to click on links to publications, links are collected in a list together with the corresponding journals, for each of which separate functions are created for data collection, since the information resources of each journal have a different structure. For correct data collection, each log must be associated with its own data collection method. After establishing the necessary links, the agent program directly collects data about each publication.

Thus, during the implementation of the agent program, functions for collecting links to publications of each country, a function for clicking on links from the list, and functions for collecting data from the pages of a particular journal are written.

2. Data storage system

The data collected by the agent is stored in a relational database. PostgreSQL is used as a database management system.

To analyze the publication activity of the BRICS countries, you need to know: how many publications did each country participate in; what organizations and authors of the country participated in what publications; what authors are associated with what organizations; what keywords apply to each scientific publication; dates of scientific publications.

To store such data in a relational database, the following entities are used: publication; keyword; organization; author; publication-organization; publication-author; author-organization; publication-keyword.

The physical model of the database is shown in Figure 2.

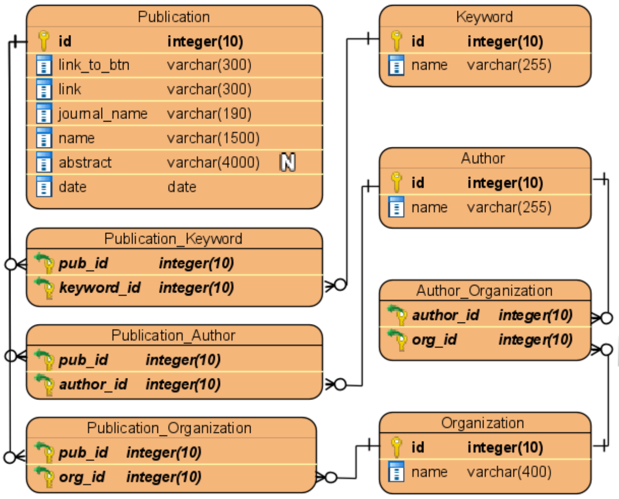


Fig. 2. Physical database model

3. Web Application

A three-level client-server architecture is selected for the web application. This architecture implies the presence of a "thin" client application for displaying data, a server application that performs the business logic of the system, as well as a database.

For the implementation of the server application, the Python programming language and the Django web framework are selected. It is a high-level open-source web framework written in Python [2]. To implement the client application, the JavaScript programming language and the React library are selected. JavaScript is a multi-paradigm programming language with dynamic typing and automatic memory management [3].

There are two actors in the system: the administrator and the user. The user should be able to view the statistics, and the administrator should be able to view them and start the process of updating them by the agent program. To start the system update process, the administrator must be authenticated in the system.

A server application is a single project with connected and disconnected applications. The BricsAgentApplication application is responsible for working with the data in it, and the AuthenticationApplication application is responsible for authentication. The software agent is part of the BricsAgentApplication application.

To implement the above features of actors, it is described how to get data in a file views.py BricsAgentApplication, which is a set of functions that are called when the client application is re-quested (the matching of the functions and their urns is described in the application file urls.py). The set of these functions is the API of the server application, its main functions are: start collecting publication data; getting the progress of the publication data collection; get a list of ten organizations in the country with a certain substring in the name and (optionally) a certain number of articles; getting the most frequently used keywords in the country's publications; get a list of the 10 authors with the most publications for a particular organization; getting the publication activity of countries; obtaining data on pairwise cooperation between countries.

The client application sends requests to the server application, which returns the necessary data by calling these functions. A client application is a Single Page Application consisting of a single page with components that replace each other. The components display certain data, according to the use cases for actors. The main class of the App application contains the code responsible for drawing all the components that replace each other in the same country.

Each component, including App, is a class that inherits the Component class provided by the Re-act library. Each component also contains functions for accessing the server for specific data. The list of components includes: a component for viewing the publication activity of countries (see Fig. 3), a component for viewing the organizations of the country with the largest number of publications, as well as with the most common keywords in publications; a component for viewing all the organizations of the country (see Fig. 4);

a component for viewing the authors of a particular organization with the largest number of publications (see Fig. 4); component for authentication; component for starting data collection, component for viewing data on pairwise cooperation of countries.

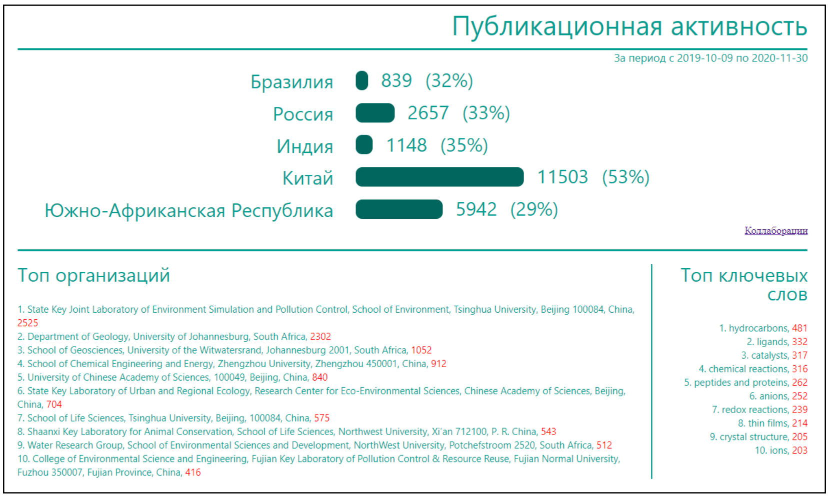


Fig. 3. Component for viewing all countries publication activities

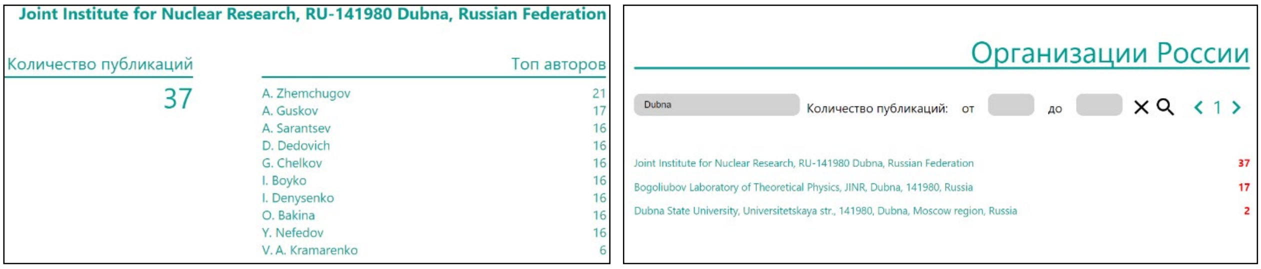


Fig. 4. Component for viewing organization information (left) and all countries organizations (right)

Conclusion

Thus, in the course of the work, data from publications of the BRICS countries were collected and web applications were created to display them and analyze the publication activity of the countries of this union.

In the future, we plan to improve the design of the client application components, as well as de-ploy the web application on a virtual machine with a static IP address provided by the Cloud service (Joint Institute for Nuclear Research).

References

1. Database of scientific publications Nature Index / Electron. Dan. – [Electronic resource]. URL: https://natureindex.com.
2. Dronov V.A. Django 2.1. The practice of creating websites in Python. — SPb .: BHV-Petersburg, 2019.
3. Haverbeke M. Expressive JavaScript. Modern web programming. 3rd ed. — SPb.: Peter, 2019.

COLLECTION OF DATA ON PASSING SCORES
IN HIGHER EDUCATIONAL INSTITUTIONS OF THE RUSSIAN FEDERATION

Sasin Alexey[1], Priakhina Daria[2]

[1] *Student;*
Dubna State University;
The International School of Information Technologies
"Data Science";
The main educational program:
Information Systems and Technologies, group 3282;
e-mail: sad.18@uni-dubna.ru.

[2] *Researcher;*
Meshcheryakov Laboratory of Information Technology;
Joint Institute for Nuclear Research.
Senior Teacher;
Department of Distributed Computer Systems;
Dubna State University.

Keywords: data collection, data extraction from web pages, multithreaded algorithms.

Introduction

After receiving secondary education, school graduates face the question of choosing an educational institution for higher or secondary specialized education. There are resources on the Internet that contain information about passing scores to higher educational institutions (universities). Examples include the site "Ucheba.ru" [1] and "Go online" [2]. The main disadvantages of such resources are: the high cost of posting information and the need for self-registration of universities. Free posting of information on such sites does not guarantee the reliability of this data. Thus, there is a need to develop a service that will provide such information about all higher and secondary specialized educational institutions of the Russian Federation, such as areas of study, passing scores, tuition fees, announcements of events for students and much more. However, this information must be reliable, therefore it is necessary to obtain it from trusted sources, which are documents located on the official websites of educational institutions.

The objective of this work is to develop an algorithm for collecting orders for admission to universities in the Russian Federation in order to further extract information on passing scores.

1. Extraction of information about universities

Information about educational institutions was obtained from the official website of the Federal Service for Supervision in Education and Science, where it is presented as an xml file [3]. This file contains information about 922 educational institutions, of which: 492 head institutions and 430 branches. From the file with data about educational institutions, their names and links to official sites were extracted using a program written in the Python programming language [4] using the ElementTree library [5]. The received data was saved to a JSON file (see Fig. 1).

```
Name,Links
Московский городской университет управления Правительства Москвы,http://mguu.ru/
города Москвы Московский городской педагогический университет,http://mgpu.ru/
города Москвы Московский государственный институт индустрии туризма имени Ю.А.Сенкевича,http://mgiit.ru/
Московской области Технологический университет,http://unitech-mo.ru
Московской области Университет Дубна,http://uni-dubna.ru
```

Fig. 1. Data on educational institutions

2. Collecting enrollment orders

The program for collecting orders for admission to universities was developed in the Python programming language [4] using the selenium framework for automated browser management [6]. Since the path to orders on the website of each university is unique, it is necessary to use a search engine, for example, "Yandex" to find a page on which the necessary information may be located. For the search, a query is set, which includes the name of the university, the year and the phrase "order of enrollment". University names are extracted from a JSON file (see Fig. 1). After receiving links from the search engine, the algorithm shown in Figure 2 is launched to collect enrollment orders.

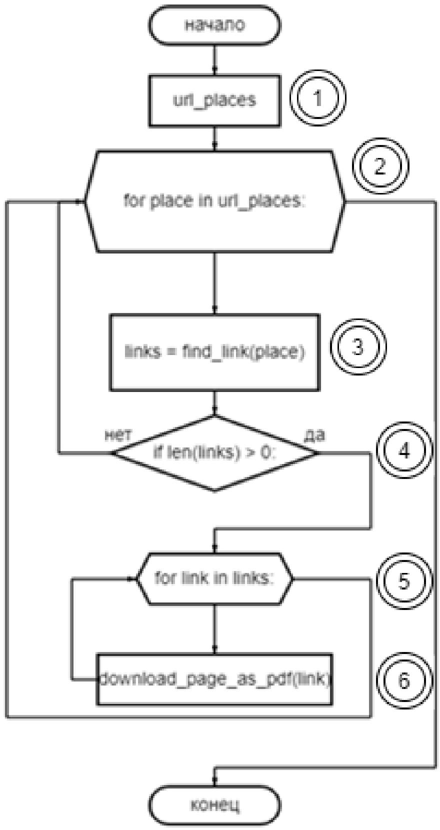


Fig. 2. Algorithm for collecting enrollment orders

The operation of the algorithm can be described as follows.

1. Clicking on each link received from the search engine. On the page that opens, a search and recording of all links is started, where the expressions are found: "enrolled", "Enrolled", "ENROLLED".
2. Loop through all links from the list formed in paragraph 1.
3. Writing to a new list of all links that were in these places or below the structural tree of the web page.
4. Checking the length of the new list for links to download documents.
5. Loop through all received links.
6. Extracting .pdf files from all links using the selenium framework [6].

The disadvantages of this algorithm include the following.

- Binding to keywords ("enrolled", "enrolled", "enrolled").
- Lack of checking the url of the site received in the search engine for compliance with the official site of the educational institution.
- Consecutive enumeration of all educational institutions.
- Data collection time is 48 hours.

Due to the above-described shortcomings of the first algorithm, it became necessary to develop an optimized algorithm (see Fig. 3).

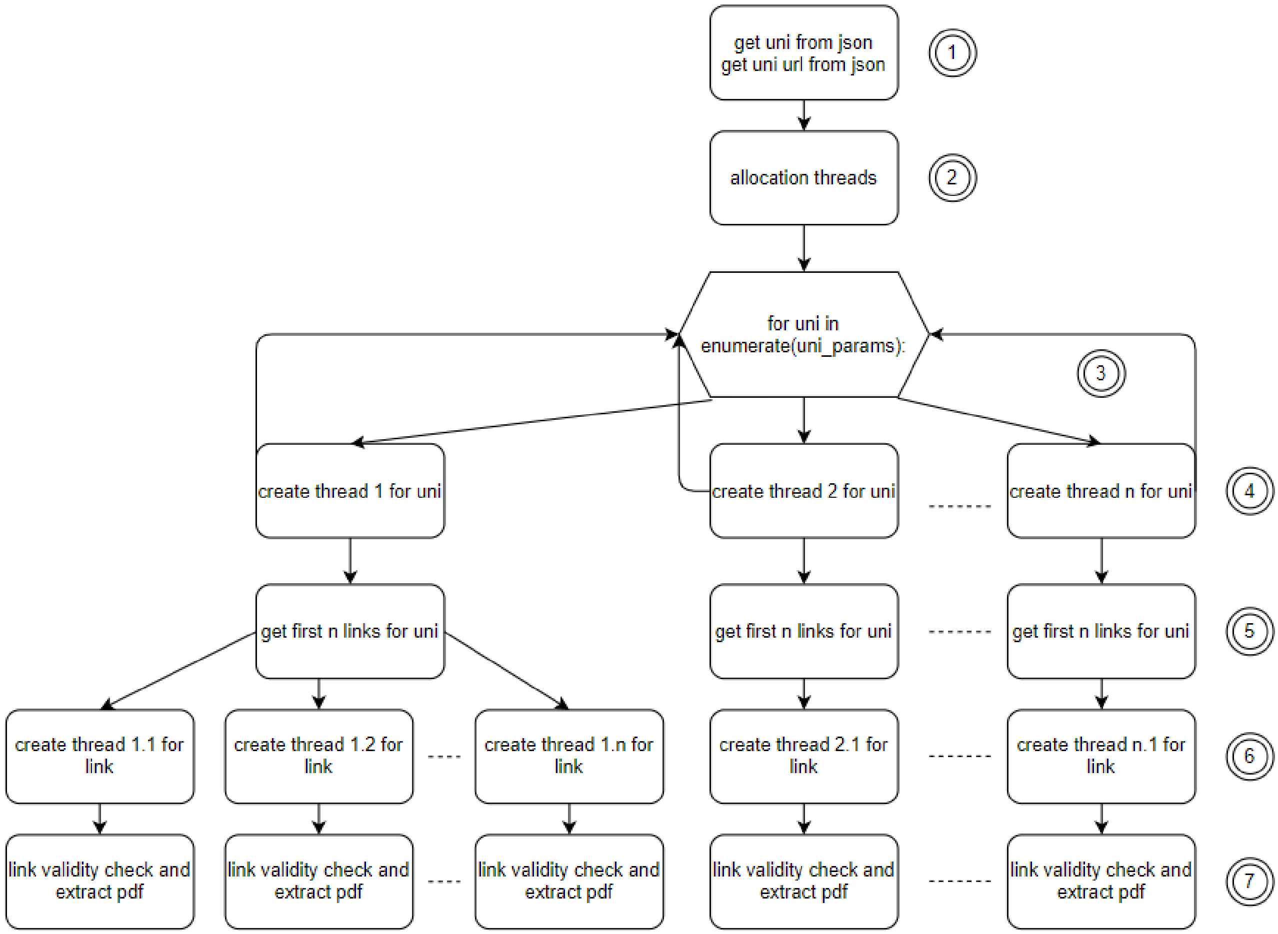


Fig. 3. Optimized algorithm for collecting orders for crediting

The optimized algorithm can be described as follows.

1. Extraction of names and links of educational institutions from a JSON file.
2. Setting the maximum number of threads (12) using the threading module.
3. Loop through all the link names.
4. A separate stream for each educational institution.
5. Getting the first N links from the Yandex search engine.
6. Loop through all the links received from the search engine with the allocation of a separate stream for each link.
7. Checking the relevance of the received links (comparing the url of the link and the url of the university) and then extracting .pdf files using the selenium framework [6].

It is also worth noting that the running time of the algorithm was 6 hours.

3. Data collection results

With the help of an optimized algorithm, orders for enrollment in pdf format for 2020 in 922 educational institutions, of which 492 are head institutions and 430 branches, were received and saved to a local computer. Each file is saved in a directory whose name coincides with the name of the university.

Conclusion

As a result of the work, two algorithms were developed for collecting orders for admission to universities in the Russian Federation, which are posted on the official websites of educational institutions, in order to further extract information on passing scores. Testing, which was carried out on a computer with an AMD RYZEN 5 3600X processor and a network bandwidth of 100 Mbps, showed that the first algorithm collects enrollment orders in 48 hours, and the optimized algorithm in 6 hours. Therefore, the optimized algorithm works 8 times faster and does not collect unnecessary data by checking the relevance of the link. Future plans include developing methods for extracting data from collected documents.

References

1. Ucheba.ru. – [Electronic resource]. URL: https://www.ucheba.ru/.
2. Go online. – [Electronic resource]. URL: https://postupi.online/.
3. Federal Service for Supervision in Education and Science. – [Electronic resource]. URL: http://obrnadzor.gov.ru/otkrytoe-pravitelstvo/opendata/7701537808-raoo/.
4. Dawson, M. Programming in Python. – SPb: Peter, 2016.
5. Official documentation of the ElementTree library. – [Electronic resource]. URL: https://docs.python.org/3/library/xml.etree.elementtree.html.
6. Avasarala S. Selenium WebDriver Practical Guide. – Packt Publishing, 2014.

INTELLIGENT NOISE REDUCTION

Rybakina Alena[1], **Klimanova Darya**[2]

[1] *Student;*
Dubna State University;
The International School of Information Technologies
“Data Science”;
The main educational program:
Computer Science and Engineering, group 4015;
e-mail: lena.ribackina@yandex.ru.

[2] *System Analyst at the Machine Learning Department;*
MTS Artificial Intelligence Center.

Keywords: noise reduction, convolutional networks, deep learning.

Introduction

During the pandemic remote collaboration tools have become a necessary means for organizing workflow. Audio calls with good and excellent speech quality are so necessary at this time. Such calls are especially in demand at meetings in large corporations, for organizing classes during the educational process in schools, universities, at entertainment events, for example, when broadcasting a symphony orchestra performance, etc. However, the environment where an audio device that receives incoming signals can be located reproduces and perceives various types of background noise. The participants of audio calls themselves cannot influence in such a way that external noise does not distort the transmitted signal. For example, these are noises such as the barking of a dog, the crying of a child, the noise of an air conditioner, traffic, the noise of cutlery in the kitchen, etc.

The purpose of speech enhancement is to use algorithms to improve the intelligibility and quality of a noise speech signal.

The development of noise reduction algorithms is complicated by such factors as: low signal-to-noise ratio, rapidly changing probability distributions, nonlinear combinations of different types of noise, etc. In such unfavorable conditions, existing noise reduction approaches can improve the signal-to-noise ratio, but, unfortunately, they will generate new distortions: unnatural speech distortions, fluctuating residual noise.

Therefore, the task of developing an algorithm for effective noise reduction on audio signals remains an urgent problem and a research topic, which is still considered in many articles.

In the 80s, this problem was solved using classical speech improvement algorithms, such as spectral subtraction, Wiener filtering, and algorithms based on statistical models.

These techniques are based on the calculation of the spectral estimate and rely on the spectral estimate of noise, which, in turn, works using the voice activity detector (VAD). Each of the three algorithms requires a careful adjustment of the parameters, and these parameters are difficult to adjust.

Neural networks, unlike probabilistic models, support the flow of information in only one direction, from input to output.

The object of the study is the study of methods that are used to solve noise reduction problems, the subject of the study is the algorithms of neural networks. Such algorithms of recurrent neural network, generative-adversarial neural network, convolutional neural networks were investigated in the work.

1. Overview of algorithms for noise reduction

The study consisted of several stages: a review of existing methods for solving the problem, data collection and preparation, and training of the selected neural network model.

The stage of reviewing the existing methods for solving the noise reduction problem was to study articles describing common architectural solutions in the field of deep learning.

The first model is based on a recurrent neural network, where the main part of noise reduction is performed on a low-resolution spectral envelope using improvement coefficients. These coefficients are calculated using a recurrent neural network. The improvement coefficients are obtained by calculating the square root of the ideal ratio mask (IRM).

This approach has a limitation – it is impossible to model smaller details in the spectrum. The solution is a comb-pitch filter to suppress inter-harmonic noise. As the frequency of the speech signal depends strongly on the frequency, the comb filter operates in the frequency domain based on the bandpass filtering coefficient [1].

The following algorithm is based on a generative-adversarial model. Generative models learn to match samples z from some prior distribution of Z, with samples x from another distribution of X. X is a data set of training examples (for example, images, audio, etc.). The component in the GAN structure that performs the mapping is called the generator (G), and its main task is to learn an efficient mapping that can simulate the real distribution of data to generate new samples.

The way in which G is learning to do the mapping is through adversarial training, where we have another component called a discriminator (D). Usually D is a binary classification, and its input data are either real samples coming from a dataset that simulates G, or forged samples composed by G.

G can slightly adjust its output waveform at the right side of the distribution. It gets rid of noisy signals, since they are identified by D as fake. In this sense, D can be understood as a loss function for G.

In the noise reduction task, the improvement is performed by the G-network. Its inputs are a noisy speech signal x together with a latent representation of z, and the output is an improved version of x' = G (x). G is a fully convolutional neural network without fully connected layers.

The work of G consists of two stages. First, the noisy speech is encoded into the latent representation, and then the latent representation and the latent vector z are decoded into a noise-free signal. At the encoding stage, the input signal is compressed through a series of dilated convolutional layers. The compression is performed until a hidden representation is obtained. At the decoding stage, the network architecture is symmetric as described above [2].

The third model under consideration differs from the previous ones in that it does not use time-frequency representations or amplitude spectrograms as input data to reduce the large dimension of the raw signals. Thus, it does not face the disadvantages associated with the rejection of potentially valuable information (in phase) and the disadvantages of using universal methods for extracting features (for example, the analysis of amplitude spectrograms). The model specifically studies the characteristics of the signal given to it.

The model architecture is also based on convolutions. This makes it flexible in the time dimension. Notably, the model uses contextual information to process audio. In order to process a single fragment of an audio signal, the model also looks at the parts of the audio recording before and after that fragment. This allows us to collect useful information for further processing of the current fragments under consideration [3].

The fourth model is TCNN. It is a fully convolutional neural network (CNN) for speech enhancement, running in real time. The model architecture is based on an encoder-decoder with an additional time convolutional module (TCM). TCM also uses dilated layers with different pitches in the encoder block.

The decoder architecture is symmetric to the encoder architecture and has skip connections to the encoder layers.

The encoder accepts a sequence of noisy frames as input. The first layer in the encoder increases the number of channels from 1 to 16. The next seven layers compress the audio size using the usual convolutions with a step size of 2.

The encoder output is converted to a one-dimensional signal. This signal is fed to the input of the TCM unit. The TCM consists of three blocks. The block in turn consists of six dilated convolutions, which have dimensions of: 1, 2, 4, 8, 16 and 32. This allows the algorithm to cover a large area of audio and extract features from an ever-increasing area under consideration. Thus, it receives global information about the received audio signal.

That is, first, the first convolution received local features of the audio signal, the second convolution with a step of two, takes the area wider (twice as many samples as the first and increases the step size twice as large as the first, i.e. considers neighboring samples, compared to the previous layer), respectively, receives information larger than the first. Thus, the most general information about the audio signal will be collected in the last layer, at the end of the block [4].

The latest DEMUCS model also consists of a convolutional encoder and decoder and a recurrent neural network between them (LSTM) with four layers. It is worth adding that the model is based on the idea of the U-net architecture, which is most often used in image segmentation tasks.

The encoder accepts the raw signal as an input and outputs its hidden representation. Next, the new representation of the signal goes through processing in convolutional layers. This representation is then fed to the input by the LSTM, which, after processing, transmits the new representation to the decoder network. After processing, the decoder outputs a clean signal. By structure, the decoder is the symmetry of the encoder and has skip connections [5].

2. Train model

The second part of the study was to prepare data for training a neural network.

Russian Open Speech To Text [6] is a large open corpus of the oral Russian speech, which contains 20 thousand hours of oral speech, from different subject areas – calls, youtube, lectures, phone conversations, books. Only audio books were used to develop the data set, since the recordings were without noise and quite a lot of voices read the books. Since the audio here varies in duration from less than a second to 17 minutes, a function was written to cut the audio and save it to a folder. No files were taken for less than 2 seconds. The cutting itself was performed for 2 seconds.

Next, a noise search was performed. 200 types of noise were formulated, where 5 audio files were found for each type. The audio was found in the Internet [7].

All these noises can be divided into the following classes: audio files in the form of popping, crunching, hissing, chewing, bubbling, tearing, gurgling, screeching, short noise, breaking objects, creaking, noises related to water – dripping, coastal, rustling, musical, knocking, noises made by people – coughing, laughing, blowing your nose, babbling, noises from equipment, ringing noises – from sirens, alarm clocks, noises that relate to the place (environment) – in a stadium, with a cafe, restaurant etc. For augmentation-applying noise to audio recordings, a pipeline was developed using the Apache Beam library. Here, the task of streaming was implemented in some circles, such a process is known as ETL, i.e., extracting, converting and loading information.

The steps of the written pipeline consisted of reading a tabular data set with a description of the clean files stored in the format .csv, then overlay the noise and write to the memory of the noisy file. Noise is selected by declaring an object of the Provider class. In the class itself, a table file is read in the format .csv, and from the resulting list, one random file is selected that will participate in the noise of the audio signal. For a noisy file, its characteristics are saved, which include the path where it is stored, the path of the file with pure speech, the path of the file with noise, the average power and the ratio of the pure signal to noise, since it can be random from -5 to 5. The last step is to record the noisy file.

To ensure that the number of noisy files by type is uniform, the code was modified so that it divides the total data set into parts according to the entered number of noise types into separate data sets and saves them in .csv format. Next, there is a search of files with noise and the augmentation process takes place as well, only in the process a certain file is selected, and after the pipeline is processed, all the files are collected.

The resulting data sets were tested on all the models described above, and Demoucs was chosen as a further step in selecting the model configuration, since it took less time to train the model and suppressed noise at most frequencies. As a quality metric, such metrics as MOS and STOI were used.

The quality of the audio files was mainly conducted by listening to audio from different classes and scoring from 1 to 5 on the parameters of speech intelligibility, noise audibility, and speech quality, since subjective metrics did not always reflect the correct assessment. It was noticed that if the number of output STOI metrics is less than 0.8, then the audio quality score for us was low (3 points). If it is more than 0.8, then we gave a high rating. If the score was no more than 0.6, then the quality was very poor (1 or 2 points). In general, bell noises or sirens always showed poor results, while the other types had consistently good results.

The project itself and its results are presented here [8].

Conclusion

During the research work, several architectures and approaches for solving adaptive noise reduction problems were considered, but also the principles and methods for solving various problems with data streaming were studied. The optimized Demoucs model can be further used in the development of a product for noise reduction during conferences, calls, both from an Android device and from a computer. We also studied the possibilities of representing audio files in the form of spectrograms, which, when compared, showed the frequencies that the neural network for one reason or another passes and removes noise in them. During the study, the Fourier algorithm, the capabilities of the libros, pandas, and matplotlib libraries were studied.

Working with the Data Flow technology clearly showed a fast and convenient process of data augmentation.

References

1. Jean-Marc Valin. A Hybrid DSP/Deep Learning Approach to Real-Time Full-Band Speech Enhancement // IEEE Multimedia Signal Processing (MMSP) Workshop, 2018.
2. Pascual S., Bonafonte A., Serra J. SEGAN: Speech Enhancement Generative Adversarial Network // Cornel University, 2017.
3. Rethage D., Pons J., Serra X. A WaveNet for Speech Denoising // Cornel University, 2017.
4. Pandey A., Wang D.. TCNN: temporal convolutional neural network for real-time speech enhancement in the time domain // ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019.
5. Defossez A., Synnaeve G., Adi Y. Real Time Speech Enhancement in the Waveform Domain // Cornel University, 2020.
6. Russian Open Speech To Text. – [Electronic resource]. URL: https://azure.microsoft.com/en-us/services/open-datasets/catalog/open-speech-to-text/#AzureNotebooks.
7. ZVUKOGRAM. – [Electronic resource]. URL: https://zvukogram.com.
8. Intelligent-noise-reduction. – [Electronic resource]. URL: https://github.com/Alena0704/intelligent-noise-reduction.

RESEARCH OF METHODS OF A PERSON BY VOICE IDENTIFICATION

Nechushkin Kirill[1], Streltsova Oksana[2]

[1] *Student;*
Dubna State University;
The International School of Information Technologies
“Data Science”;
The main educational program:
Computer Science and Engineering, group 4013;
e-mail: nka.17@uni-dubna.ru.

[2] *Candidate of Physics and Mathematics, Senior Researcher;*
Meshcheryakov Laboratory of Information Technology;
Joint Institute for Nuclear Research.
Associate Professor;
Department of Distributed Computer Systems;
Dubna State University.

Keywords: neural network, voice, recognition.

Introduction

In this paper, the subject of research is oral speech. Oral speech is individual in sound. Individual features of speech – a set of characteristics that are determined by the peculiarities of the structure of the vocal organs and individual characteristics of a person. The most striking external manifestations of individual characteristics are such characteristics as the timbre of the voice and the tempo of speech.

The main tasks in the field of speech processing are speech recognition (text-to-speech) and voice identification, which is part of the task of full biometric authentication. There are also other tasks related to audio, for example, removing noise from sound recordings or voice cloning, which allows to train the program to speak any words and sentences with any voice, data about which is available in the system.

Today there are voice biometrics solutions that are used in call centers to identify fraudsters or in the banking environment for biometrics.

The main objective is to develop an algorithm based on a neural network approach to solve the problem of identifying a person by his voice. This includes subtasks:

- creating a training dataset (including the data collection methodology, data transformation, format selection, etc.);
- design of the neural network architecture;
- researching on the applicability of the chosen architecture to solve the task.

1. Audio file

It is important to understand the structure of the audio file, when creating a neural network that works with audio. In order to train a neural network, it is necessary to create a data set, which in this case will completely consist of audio files.

Audio files are a recording of a human voice in the case of voice identification. This recording was captured using a microphone and it is originally an analog signal but it is stored digitally on the computer. This is possible by analog-to-digital audio conversion. An analog audio wave can be represented as a function. To digitize it, discretization is performed, that is, the value of the function is calculated for each argument as shown in Figure 1. In the case of an audio file, instead of an argument, we have a time value, and an amplitude value as a function value. The set of argument values (time values) is called the sampling rate, and the sampling step is the difference between the values of two adjacent arguments [1].

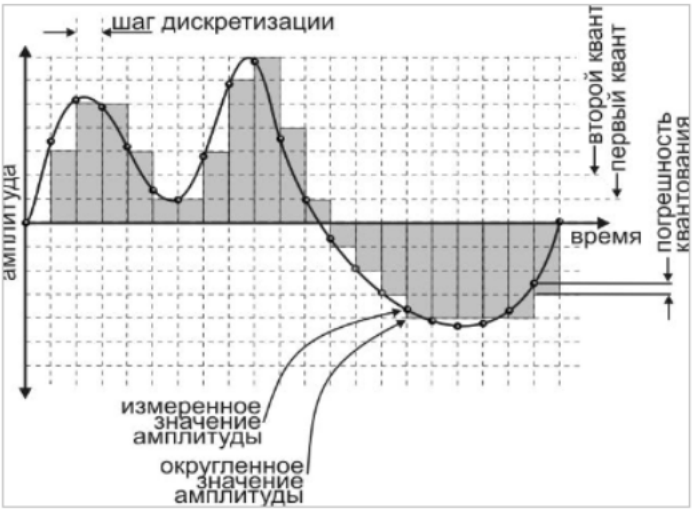


Fig. 1. Analog-to-digital sound conversion

2. Collecting data

It is necessary to collect voice data to identify a person by his voice. In this project, it was decided not to use existing data sets, but to generate new ones.

For this project, I have decided to use audio data from open sources, and then transform it as best suited for neural networks. It was experimentally found that single-channel audio with a sampling rate of 16,000 Hz, a file length of one second and the absence of pauses in speech in it show themselves better than others. Pauses in speech can be up to several seconds, and since the file length was one second, the resulting files will contain no information about the voice.

Therefore, after receiving audio data, it was necessary to remove all the pauses from the audio. Since audio files have been recorded in different conditions, it is necessary to use a method for removing speech pauses, which will dynamically determine these pauses. For this purpose we have used the pyAudioAnalysis [2] library. This library was created specifically for creating datasets for neural networks and includes many necessary methods but for this project only the method of removing silence is sufficient. You can see what this method does in Figure 2.

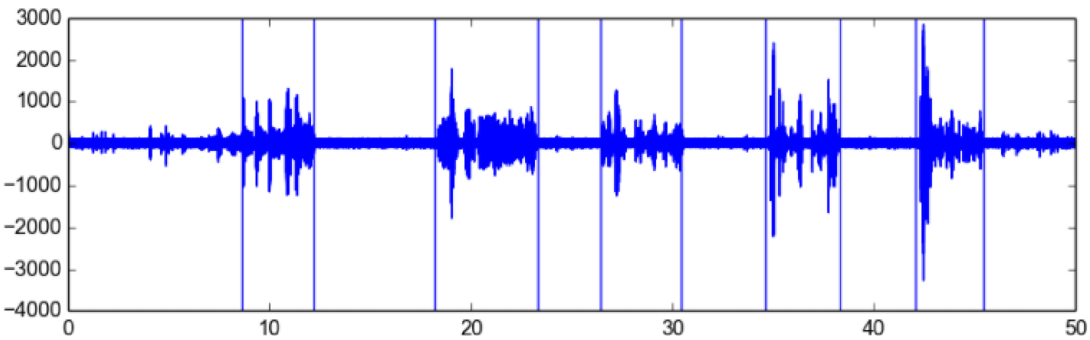


Fig. 2. SilenceRemoval method

After removing the pauses using pyAudioAnalysis library, audio files of various lengths were obtained, with which it was subsequently necessary to perform further conversion. For further operations on the audio, it was decided to use the pydub library [3], since this library contains all the tools necessary for further work: splitting an audio file, changing its sampling rate and changing the number of channels of an audio file. Using the pydub library, a class was created that allows you to split an audio file into multiple files with a given step.

To analyze the received audio data, the torchAudio library and a practical guide for working with this library were used [4]. For example, with the help of this library, it was found that before the conversion, the audio had two-channel sound and a sampling rate of 48 000.

3. Data altering

In this project I used the mel-spectrogram as a method for representing the audio file, since the mel-spectrogram better reflects the features of speech, and the spectrogram gives more information about the audio file. Mel-spectrogram is shown in Figure 3.

Mel is a unit of sound based on the perception of that sound by our hearing organs. It is also commonly used in audio analysis. The sound is perceived by the human ear non-linearly. This dependence is described by a mathematical formula: $m = 1125 \ln(1 + f/700)$.

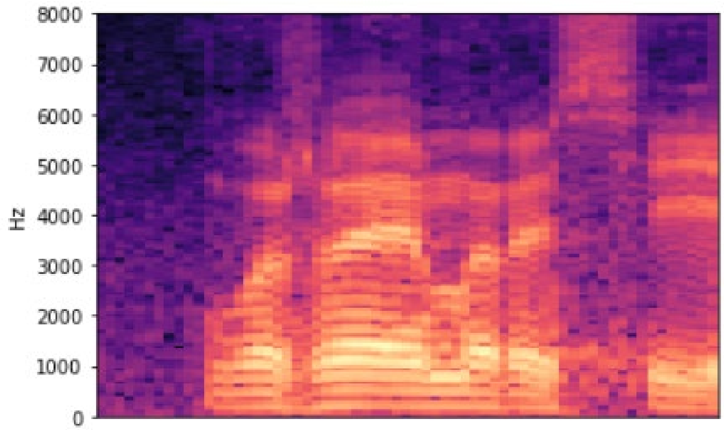


Fig. 3. Mel-spectrogram

After all, we get a ready-made dataset for training neural networks of different types and architectures. The resulting dataset was created to identify a person by voice, but can also be used for the task of voice cloning.

4. Neural network model creation

Tensorflow framework was used to create a neural network model [5]. The neural network architecture consists of five layers: an input layer, a convolutional layer, a subsampling layer by maximum value, a convolution layer, a layer that transforms a matrix into a vector, and a regular fully connected layer. You can see how most of the CNN looks like in figure 4. The output is a number that represents the number of the speaker who owns the voice from the audio file.

You can not be limited in neural network architecture, if you are using spectrograms because a spectrogram is very similar to black-and-white images in its structure, in contrast to audio in its original form, which is more like a one-dimensional array [6].

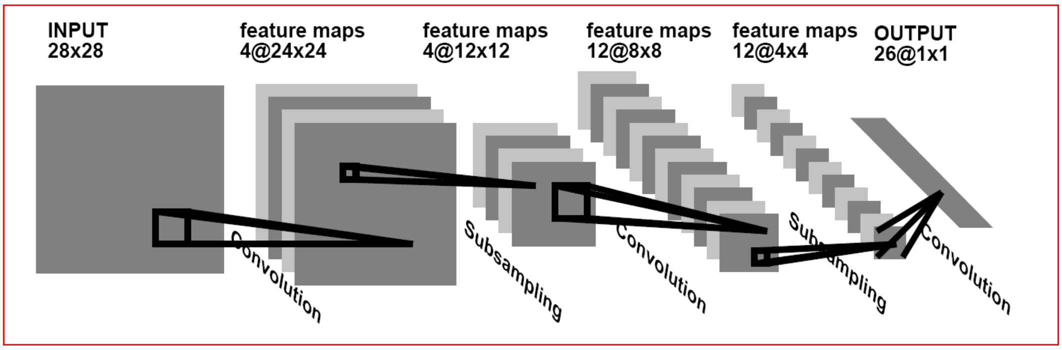


Fig. 4. Convolutional neural network architecture

Conclusion

The paper presents the initial stage of the study of the capabilities of neural network models in the problem of voice identification. A trial training sample has been collected, containing voice samples of two speakers. The work is the first stage in the study of the possibilities of a neural network approach to the problem of voice identification and the formation of a workflow for research of this kind. As part of further research, more people's voices are being added to the training sample and the corresponding improvements to the audio data analysis scripts are being made.

At the moment, using a convolutional neural network, an accuracy of 99.9% has been achieved for recognizing two people. As a trial result, such efficiency is a sign that the chosen research direction is interesting enough for further study.

References

1. Shchyogolev A.V. Recognition using neural networks sound. Course work // Kuban State University, 2018.
2. PyAudioAnalysis library documentation. – 2019. – [Electronic resource]. URL: https://github.com/tyiannak/pyAudioAnalysis.
3. James R. Pydub library documentation. – 2016. – [Electronic resource]. URL: https://github.com/tyiannak/pyAudioAnalysishttps://github.com/jiaaro/pydub/releases.
4. TorchAudio beginner's guide. – [Electronic resource]. URL: https://pytorch.org/tutorials/beginner/audio_preprocessing_tutorial.html.
5. Frameowk tensorflow. – [Electronic resource]. URL: https://www.tensorflow.org/.
6. LeCun Y., Bengio Y. Convolutional Networks for Images, Speech, and Time-Series, in Arbib, M. A., The Handbook of Brain Theory and Neural Networks. – MIT Press, 1995.

ROAD MARKING RECOGNITION AND TRAFFIC ANALYSIS

Melnik Nikolay[1], Streltsova Oksana[2]

[1] *Student;*
Dubna State University;
The International School of Information Technologies
"Data Science";
The main educational program:
Computer Science and Engineering, group 4011;
e-mail: columbiysky@uni-dubna.ru.

[2] *Candidate of Physics and Mathematics, Senior Researcher;*
Meshcheryakov Laboratory of Information Technology;
Joint Institute for Nuclear Research.
Associate Professor;
Department of Distributed Computer Systems;
Dubna State University.

Keywords: computer vision, deep learning, neural networks.

Introduction

This article discusses the possibility of analyzing the traffic situation using computer vision, which includes the recognition of objects and road users. The task of this work is to study computer vision technologies to recognize road users, such as pedestrians, cars, road signs, traffic lights, using neural networks.

1. Selecting a data set

There are many datasets in the Internet, containing marked pictures of cars, road signs, etc. The BDD100k dataset was chosen because it contains all the necessary information and is quite popular in the community, so there is enough information on the network on how to work with this dataset. Example (see Fig. 1) cars in blue, pedestrians in red, traffic lights in yellow and road signs in purple.

Fig. 1. An example of marking objects in a photo in the dataset

2. Selecting the means of implementation

When implementing this work we used: the programming language Python, the computer vision library OpenCV, deep learning framework Darknet. The architecture of the neural network is YOLOv3.

Initially there was an attempt to use YOLOv4 architecture. To train the network on the chosing dataset it is necessary in the configuration file to change the value of the attribute classes to the number of recognized objects, in this work it is 10. Next, in all the layers that come before those in which you have changed the attribute classes need to change the max_batches, which is equal to the multiplication of the number of classes: 20000. But after training the network recognition turned out to be inaccurate, many objects were not recognized at all, and those that were recognized the network marked in the wrong place. When using YOLOv3 such a problem did not arise.

3. Neural network training

Google Colaboratory was chosen as a working environment, due to the ease of installing the necessary libraries. Before starting training it is necessary to prepare the data, namely to create a .names file filled with the names of the classes of recognized objects. .train and .val files containing paths to the photos from the training and validation sets. .txt files for each photo from the training set, the files contain information about the class of the object on the image, and its position. .data file containing information about the number of classes from the .names file paths to .train, .val, .names files and path to the directory where the current neural network configuration will be saved to. (see Fig. 2) [1].

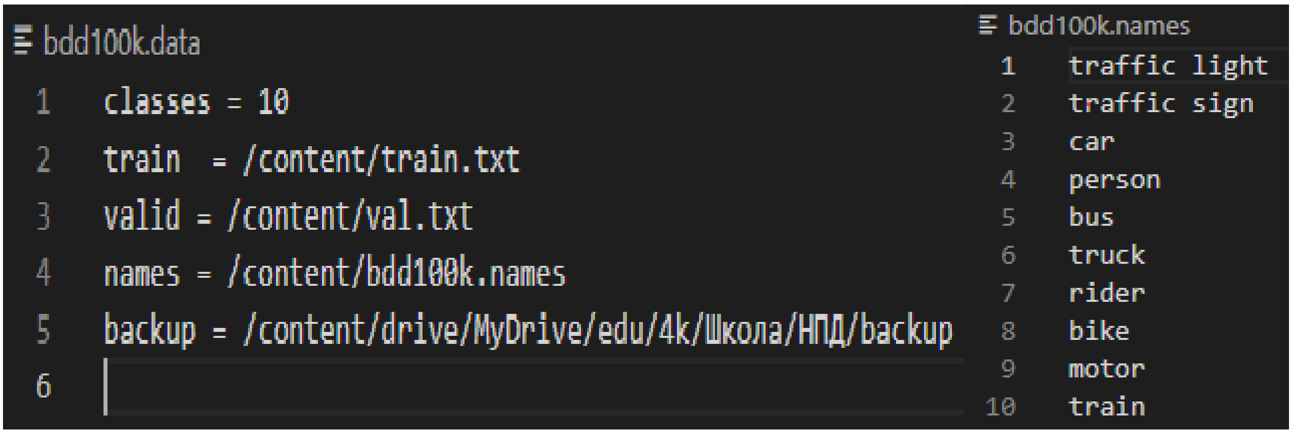


Fig. 1. Completed .data and .names files

Training of the neural network is started with the command “!./darknet detector train *.data *.cfg weights -dont_show” Training lasted about two hours, 18671 epochs of training were passed, the average recognition error was 7.391626 [2].

4. Learning outcomes

If you recognize an image from the test set, then 15 of the 16 objects in the image will be recognized correctly (see Fig. 3).

Fig. 2. Recognition result using a photo from the test set

When using a random picture from the Internet, the result is also satisfactory 8 correctly recognized objects out of 13 (see Fig. 4).

Fig. 3. Recognition result using a photo from the Internet

When recognizing objects on video, the result is slightly worse, about 60% of correctly recognized objects in the frame, with an average processing rate of 33 frames per second.

Conclusion

This work is the initial stage of the study, and the results obtained are of an evaluative nature. In the course of further research, one of the following stages is the stage of obtaining a statistically reliable assessment of the effectiveness of the obtained solution. Also, one of the areas that should be noted is the development of an algorithm for recognizing road markings and segmentation of traffic lanes, which can have a significant impact on the final efficiency of solving the problem. Also, the development plans to investigate the effectiveness of the ensemble of neural networks.

References

1. The official documentation for the Darknet framework. – [Electronic resource]. URL: https://github.com/AlexeyAB/darknet#how-to-train-to-detect-your-custom-objects.
2. Schollet F. Deep Learning in Python. – SPb.: Peter, 2018.

DEVELOPMENT OF A SERVICE BASED ON A NEURAL NETWORK APPROACH TO TRANSFORM AN IMAGE INTO AN ANIMA CHARACTER

Bachinsky Stanislav1, Kolobov Sergey[2], Streltsova Oksana[3]

[1] *Student;*
Dubna State University;
The International School of Information Technologies
"Data Science";
The main educational program:
Automation of Technological Processes and Production, group 4231;
e-mail: staz26@yandex.ru

[2] *Student;*
Dubna State University;
The International School of Information Technologies
"Data Science";
The main educational program:
Software Engineering, group 4251;
e-mail: s.kolobov99@mail.ru

[2] *Candidate of Physics and Mathematics, Senior Researcher;*
Meshcheryakov Laboratory of Information Technology;
Joint Institute for Nuclear Research.
Associate Professor;
Department of Distributed Computer Systems;
Dubna State University.

Keywords: anime, selfie, style transfer, machine learning, neural networks.

Introduction

In the modern world, applications for processing and styling images obtained when photographing with a digital camera are very popular: all kinds of filters, masks, as well as representations of one image in the style of another.

This article discusses methods of transferring a style from one image to another to solve the problem of synthesizing an image of an anime character, based on an image of a real person, to further create a service for converting an image of a person into an anime character.

1. VGG-19 Neural network architecture

Transferring a style from one image to another can be considered a texture transfer problem. When transferring a texture, the goal is to synthesize the texture of the source image so as to preserve the semantic content of the target image [1].

To transfer styles, we use the VGG-19 network, which consists of a number of convolutional layers, pooling layers and several fully connected layers. VGG-19 is a neural network capable of classifying over a thousand different objects. This network has been trained on the huge ImageNet dataset. On each of its layers, the neural network extracts some elements from the image that represent its characteristics. In the end, all the characteristics add up to the final classification.

The VGG-19 is divided into two parts:

- VGG-19 feachures, which represent all convolutional and downsampling layers;
- VGG-19 classifier - three linear layers of the classifier.

There are use only a fraction of the convolutional and downsampling layers of the pretrained VGG-19 model (see Fig. 1).

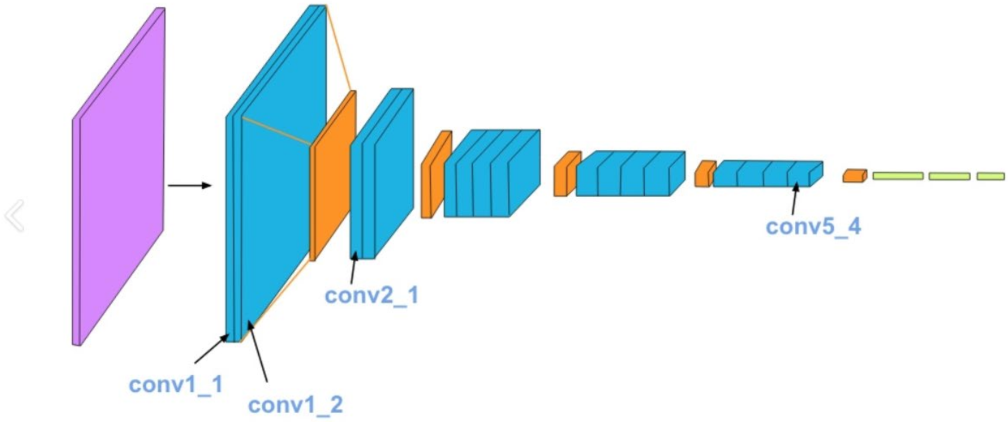


Fig. 1. The structure of the neural network VGG-19

The resulting image is obtained in an iterative manner. At each iteration, we feed the current image to the input of the neural network and obtain a set of features for it. We calculate the losses from the original image and the stylistic image. Using the resulting losses and their corresponding weights, calculate the total loss and use any gradient descent optimizer to modify the generated image so that it reduces its loss after each iteration (see Fig. 2).

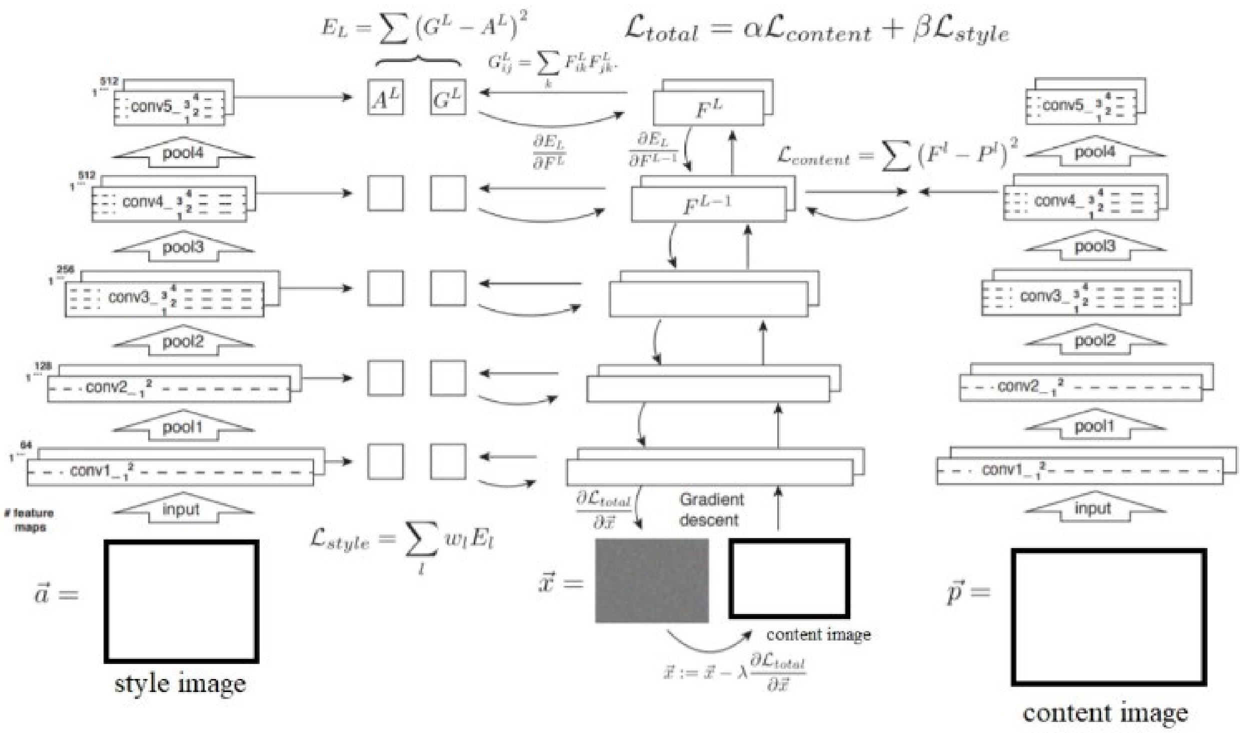


Fig. 2. The process of obtaining the resulting image

2. Cross domain style transfer with CycleGAN

CycleGAN is a type of generating network used to convey a Picture Style. Learning is done without a teacher, which means that it is impossible to unambiguously match images from both of these domains. The network is able to recognize objects in the images of the source domain and perform the necessary transformations to match the appearance of the object in the images of the target domain. The original implementation of this algorithm was trained to "turn" horses into zebras, apples into oranges, and photographs into paintings. Using images of human faces taken from the Selfie2Anime dataset [2], we will train a pair of generative adversarial networks, one studying the visual style of selfies and the other studying anime.

The learning process begins with an original image of a person's face. We will train two deep networks, one generator and one discriminator. The discriminator will eventually learn to distinguish between real and simulated facial images. The generator will be trained to transform the input image from the source domain to the target domain using random images of anime characters from the training set [3].

To make sure that this transformation makes sense, we introduce a recovery condition. This means that we are simultaneously training another generator / discriminator set that reconstructs the image in the original domain from the target domain. We comply with the condition that this reconstruction should be "similar" to the original image by calculating the value of the loss function, which we seek to minimize in the learning process (see Fig. 3, 4). This is similar to an autoencoder, except that we are not looking for the encoding in the hidden layer for the intermediate step, but for the entire image in the target domain [4].

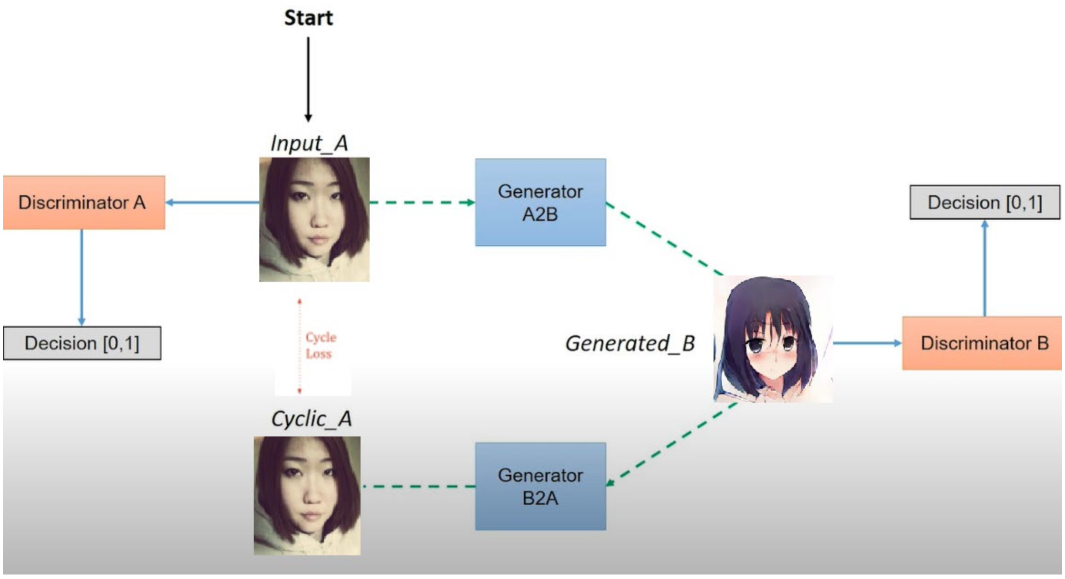


Fig. 3. Scheme of work CycleGAN network

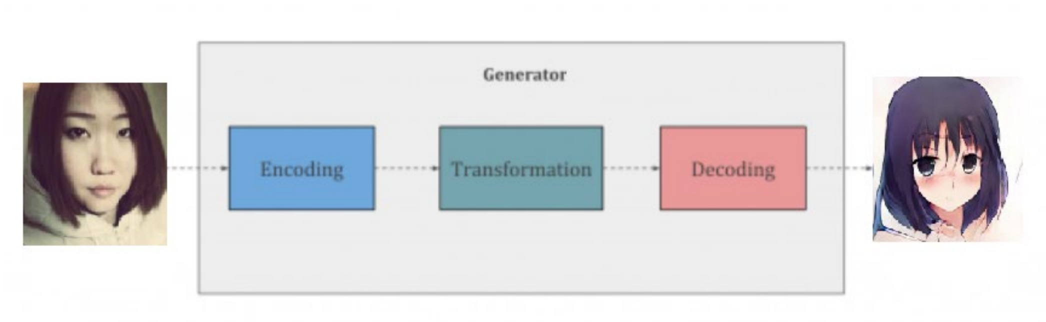


Fig. 4. The generator circuit shown in Figure 3

The generator network used consists of three main convolutional blocks. The first one finds the encoding of the image of a person's face in a hidden layer of a lower dimension. This encoding is converted to another one that represents the anime character on the same hidden layer. The decoder then creates an output image from the converted encoding, giving us an image of a human face that looks like an anime character [3]. Examples of the images obtained are shown in Figure 5.

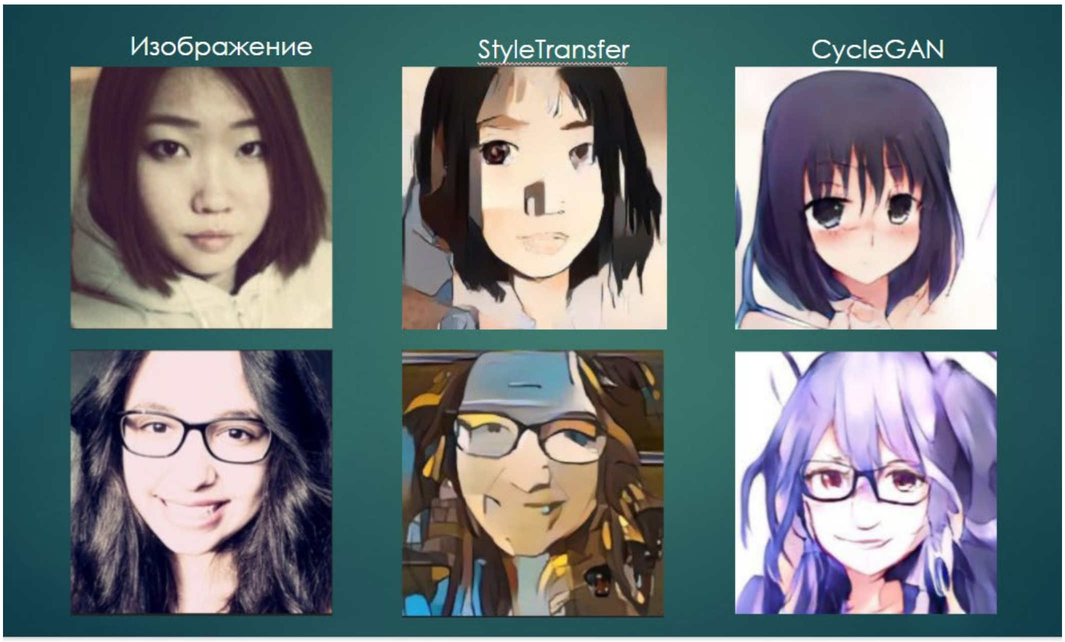


Fig. 5. Examples of received images

Conclusion

We compared the methods of transferring the image style described above. We conducted a visual expert assessment of the results obtained, since such an assessment will be applied by real users when using the application. The results of the neural networks were demonstrated to 30 independent experts to choose the best of the algorithms. According to the results of the evaluation, 100% of experts recognized the results of CycleGAN's work as the best.

In the future, it is planned to develop a service for converting a human image into an anime character using the CycleGAN neural network.

References

1. Leon A. Gatys L.A., Ecker A.S., Bethge M. Image Style Transfer Using Convolutional Neural Networks. – 2019. – [Electronic recourse]. URL: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Gatys_Image_Style_Transfer_CVPR_2016_paper.pdf.
2. Dataset *Selfie2Anime*. – 2020. – [Electronic recourse]. URL: https://www.kaggle.com/arnaud58/selfie2anime.
3. Tensorflow CycleGAN. – 2020. – [Electronic recourse] URL: https://www.tensorflow.org/tutorials/generative/cyclegan.
4. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. – 2019. – [Electronic recourse]. URL: https://junyanz.github.io/CycleGAN/.

COMPUTING & SOFTWARE FOR THE EXPERIMENTS AT THE NICA ACCELERATOR COMPLEX

ADAPTATION OF THE L1 TRACK RECONSTRUCTION METHOD OF THE CBM EXPERIMENT TO FAIR FOR THE COORDINATE DETECTORS OF THE EXPERIMENT BM@N OF THE NICA PROJECT

Khodyrev Ivan[1], Gertsenberger Konstantin[2], Priakhina Daria[3]

[1] *Student;*
Dubna State University;
The International School of Information Technologies
"Data Science";
The main educational program:
Applied Mathematics and Computer Science, group 4181;
e-mail: hodyrev.i@mail.ru.

[2] *Candidate of Technical Sciences, Head of the Group;*
Veksler and Baldin Laboratory of High Energy Physics;
Joint Institute for Nuclear Research.
Associate Professor;
Engineering Physics Institute;
Dubna State University.

[3] *Researcher;*
Meshcheryakov Laboratory of Information Technology;
Joint Institute for Nuclear Research.
Senior Teacher;
Department of Distributed Computer Systems;
Dubna State University.

Keywords: reconstruction of particle tracks, CBM L1 algorithm, cellular automata, BM@N experiment, NICA complex.

Introduction

Experiments in elementary particle physics are aimed at studying the processes that take place in experimental complexes. One of the complexes for conducting research on the properties of dense baryonic matter is the NICA complex, created on the basis of the Joint Institute for Nuclear Research.

NICA (Nuclotron-based ion collider facility) is a superconducting accelerator-storage complex of heavy ions, which has been under construction since 2016 at the V. I. Veksler and A. M. Baldin Laboratory of High Energy Physics of the Joint Institute for Nuclear Research (LHEP JINR), in the city of Dubna, Moscow Region [1, 2].

The BM@N experiment, short for Baryonic Matter at Nuclotron, is being conducted at the JINR LHEP as part of the NICA project. In this experiment, collisions of ions with a fixed target at energies up to 6 GeV per nucleon are studied, providing an opportunity to conduct research in the field of superdense nuclear matter [3].

To obtain information about the particles formed as a result of collisions, according to the data obtained from the detectors of the installation during the experiment, events are reconstructed. The result of the reconstruction is the reconstructed tracks of the particles registered by the detectors and their characteristics, for example, the mass of the particle and its momentum, as well as the primary vertex of the collision and other auxiliary characteristics of the events that occurred.

In the course of the work, the L1 particle track reconstruction algorithm used in the CBM experiment [4] at the FAIR facility [5] was adapted for the BM@N experiment. This algorithm, based on cellular automata, uses the Kalman filter method [6] at the stage of fitting particle tracks and the least squares method for evaluating the quality of the track parameters [7, 8].

1. Adaptation of the CBM L1 track reconstruction algorithm in the BM@N experiment

For the BM@N experiment, the BmnRoot software is being developed [9], based on the CERN ROOT software environment [10] and the FairRoot object-oriented framework [11]. The BmnRoot environment provides a set of classes and convenient tools for studying the performance of a BM@N installation, developing event reconstruction algorithms, and physically analyzing data.

In the BmnRoot software environment, two algorithms for the reconstruction of tracks (tracking) of charged particles are being developed simultaneously: CellAuto [9] and an algorithm based on CBM L1 tracking.

The original L1 particle track reconstruction algorithm, based on the cellular automata method, uses the recorded signals from the silicon stations of the CBM experiment. To apply it in the BM@N experiment in the BmnRoot software environment, changes were made to the stages of track reconstruction of the L1 method, such as: digitization, cluster formation, search and fitting of track candidates.

The main difference between the two experiments from the point of view of reconstruction of particle tracks is the sets of coordinate detectors that register signals from flying particles. This difference requires the preparation of a geometric configuration describing the detectors of the BM@N experiment in the structures used in the L1 algorithm. It is also necessary to form a description of the detector responses and store them in the form of objects used by the original L1 particle track reconstruction algorithm.

The first stage of the adaptation of the particle track reconstruction algorithm was the creation of a structure describing the geometric configuration of the BM@N coordinate detector stations for the 6th experiment session: silicon detectors and gas electron multipliers (GEM). Next, the detector responses are processed using the BmnRoot software environment, and the pre-processed digitized signals from the detector received during the particle's passage through the detector, called digits, are converted to the format used in the L1 particle track reconstruction algorithm. In the subsequent stages of the implemented algorithm, the geometric configuration of the coordinate detectors and the digits converted to the required format are used.

In the BmnRoot software environment, classes are implemented that represent the description of detector responses in the form of a list of digits. For the internal track system, two lists of digits are formed: from the silicon detector and from the GEM detector. For each digit, an increase in the maximum value of the signal amplitude was required. Also, the digits from both lists are converted to the format of the digits of the L1 algorithm and form a single list.

The list of converted digits from the internal track system of the BM@N experiment is then used to form clusters. The center of gravity is calculated for each cluster. The coordinate of the passage of particles through the detector station is determined by the intersection of a pair of clusters in the reading planes of the detector stations. This stage was tested by pairwise comparison of the work of two algorithms, CellAuto and L1, on the simulated data of the BM@N experiment.

The list of recovered coordinates of the passage of particles through the detector stations is then used by the track candidate search algorithm, which is based on the cellular automata method. The main concept that this algorithm uses is a triplet — three hits at neighboring stations that make up part of a possible track. At the first stage of the algorithm, all the triplets are searched for all the triplets of neighboring stations. At the second stage, they are fitted to refine the track model. To form track candidates, the triplets are combined by two matching hits. Next, the particle tracks are selected in accordance with the following conditions:

1. The longest track is selected from a pair of tracks that have shared hits.
2. For the same length of tracks in a pair with common hits, the track with the lowest value of the standard deviation is selected.

As a result of the algorithm, a list of particle tracks is created. The CbmStsTrack class stores information about the found track: a list of IDs of the recovered hits that make up the track, the particle momentum, and other auxiliary properties. For each particle track, the parameters of the track model and the error covariance matrix for these parameters are calculated.

The list of recovered particle tracks is further processed by the track fitting algorithm to improve the track model. In the L1 track reconstruction algorithm, the fitting method is implemented using the Kalman filter.

At the end of the algorithm, the results of each stage of reconstruction of L1 particle tracks are saved to a file with the .root extension. The structure of this file is a hierarchical tree, the branches of which are lists of objects obtained when performing each of the tasks of the reconstruction macro. In addition, a task was created to evaluate the performance of the new adapted algorithm, the results of which are saved as histograms in a separate ROOT file.

2. Track reconstruction quality assessment

Using the BmnRoot macro from the initial state files after the collision of particles obtained by the DCM-QGSM event generator (Dubna Cascade Model – Quark-Gluon String Model, Dubna cascade model with a quark-gluon string model), 10 thousand Monte Carlo collision events of an argon beam and a copper target at an energy of 3.2 GeV/nucleon were simulated. These events were then reconstructed using the adapted L1 method presented above.

To evaluate the effectiveness of the hit reconstruction, a histogram of the distribution of the number of recovered hits and Monte Carlo points across the GEM detector stations is constructed. This histogram is shown in Figure 1.

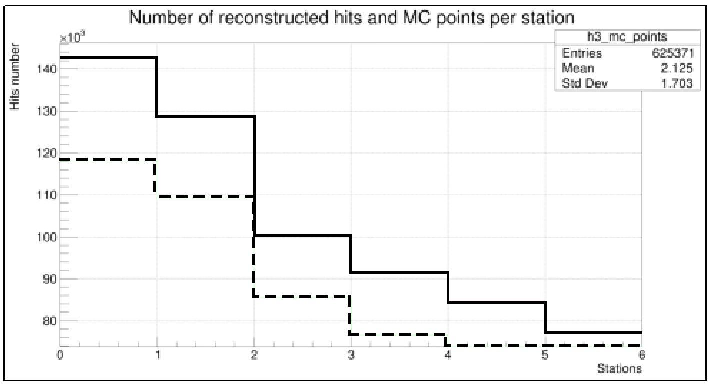


Fig. 1. Number of recovered hits (dashed line) and Monte Carlo points (solid line) by GEM detector stations

The average efficiency of the adapted algorithm for reconstructing the coordinates of the flyby of particles was about 0.86 (86%). To assess the quality of the adapted L1 particle track reconstruction algorithm, histograms of the distribution of the number of tracks in the event are constructed (see Fig. 2).

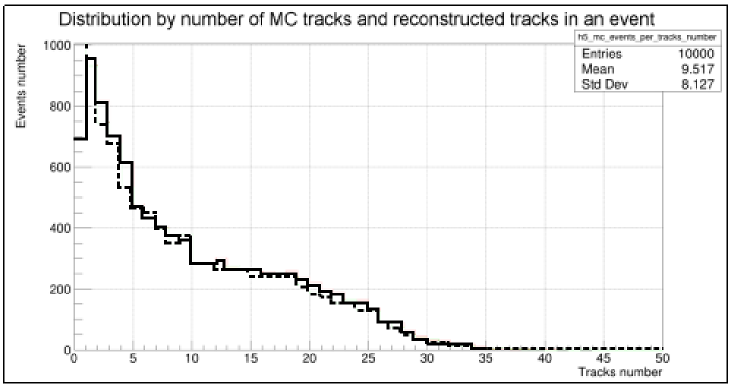


Fig. 2. Distributions of the numbers of reconstructed tracks (dashed line) and Monte-Carlo tracks (solid line) in the event

Thus, in this paper, the algorithm for reconstructing the tracks of CBM L1 particles for the BM@N experiment is adapted. For the X coordinate, the residual modulo on the test model data was 0.0065 cm, for the Y coordinate — 0.041. The efficiency of this algorithm for reconstructing particle tracks was about 85%.

The evaluation of the quality of the algorithm on the simulated data showed that the algorithm meets the specified initial criteria.

Conclusion

In the course of the work, the existing algorithm for reconstructing particle tracks in the BM@N experiment was investigated and described. The stages of reconstruction of particle tracks in high-energy physics experiments are reviewed. The methods for reconstructing the coordinates of the particle flight, selecting a track model, searching for track candidates, and fitting particle tracks, as well as their implementation in the BM@N experiment, are studied. The algorithm for reconstructing the L1 particle tracks of the CBM experiment is studied. The methods for evaluating the quality of algorithms for reconstructing particle tracks are also discussed.

In the course of the work, the CBM L1 particle track reconstruction algorithm was adapted for the BM@N experiment and the CERN ROOT environment macro was implemented to perform the reconstruction chain in accordance with the new algorithm. The quality of the adapted algorithm based on the simulated data is evaluated by the means of the BmnRoot software environment.

The efficiency of the L1 particle track reconstruction algorithm adapted to the BmnRoot software environment of the BM@N experiment for the simulated data was about 85%. This reconstruction method can be run by the task execution manager together with other algorithms of the BmnRoot software environment. The results of evaluating the quality of the algorithm stages on the simulated data are presented in the form of corresponding histograms.

The work on the adaptation of the algorithm will continue, and the correction and debugging of this algorithm will also be carried out on the experimental data of the BM@N experiment.

References

1. Kekelidze V., Kovalenko A. NICA Complex and JINR — status and plans. – [Electronic resource]. URL: https://www.researchgate.net/publication/263052453_NICA_Complex_and_JINR_-_status_and_plans.
2. Report on the status and progress of the mega-project "Complex of superconducting rings on counter beams of heavy ions" (NICA Complex). – [Electronic resource]. URL: https://nica.jinr.ru/docs/Report_NICA_SB_(12)_short.pdf.
3. BM@N Conceptual Design Report. – [Electronic resource]. URL: http://nica.jinr.ru/files/BM@N/BMN_CDR.pdf.
4. The CBM Experiment at FAIR. Volker Friesel Gesellschaft für Schwerionenforschung mbH Planckstraße 1, 64291 Darmstadt, Germany.
5. GSI FAIR. – [Electronic resource]. URL: https://www.gsi.de/en/researchaccelerators/fair.html.
6. Jose A. Hernando. The Kalman Filter Technique applied to Track Fitting in GLAST // SCIPP 98/18, 1998, University of California, Santa Cruz.
7. Nikityuk N. M. Methods of information processing from high-energy charged particle track detectors // Physics of Elementary particles and Atomic Nuclei, JINR, 1995. – Vol. 26. – Issue 3.
8. Kiryakov A. A. Methods of coordinate reconstruction in silicon microstrip detectors // IFE preprint 2003-38 ONF, Protvino, 2003.
9. Batyuk P., Gertsenberger K., Merts S., Rogachevsky O. The BmnRoot framework for experimental data processing in the BM@N experiment at NICA // EPJ Web of Conferences 214 (05027), 2019.
10. About ROOT. – [Electronic resource]. URL: https://root.cern/about/.
11. Al-Turany M., Uhlig F. FairRoot Framework. – [Electronic resource]. URL: https://pos.sissa.it/070/048/pdf.

NUMERICAL MODELING OF DIQUARKS IN DENSE AND HOT NUCLEAR MATTER

Papoyan Georgiy[1], **Kalinovsky Yuri**[2]

[1] *Student;*
Dubna State University;
The International School of Information Technologies
"Data Science";
The main educational program:
Applied Mathematics and Computer Science, group 4181;
e-mail: pgv.17@uni-dubnal.ru.

[2] *Doctor of Physics and Mathematics, Leading Researcher;*
Meshcheryakov Laboratory of Information Technology;
Joint Institute for Nuclear Research.
Professor;
Department of Higher Mathematic;
Dubna State University.

Keywords: Nambu-Jona-Lasinio, mesons, diquarks.

Introduction

Description of the properties of nuclear matter at high temperatures and densities is an urgent problem in the physics of heavy ion collisions. At present, experiments on nuclear collisions at high energies at the LHC and RHIC, as well as those planned in the near future at FAIR and NICA, will provide data on the behavior of hadrons under extreme conditions.

Quantum chromodynamics (QCD) describes the interaction of quarks and gluons based on the exchange of color charges, leading to the phenomenon of quark confinement. It is well known that direct calculations in QCD are impossible, and various effective models are used to study the properties of hot and dense nuclear matter, which reproduce all the properties of QCD at low energies. A study of the behavior of the diquark mass at a finite temperature has been carried out.

One of these models is the Nambu-Jona-Lasinio (NJL) model, which arose in the 60s of the last century with the aim of explaining the nature of chiral symmetry breaking.

The use of the Nambu-Jona-Lasinio model is convenient for describing the properties of mesons and studying the properties of hadronic matter at a finite temperature. However, the study of baryons in the NJL model is already a nontrivial task. The solution of the three-particle Fadeev equation in the NJL model is reduced to the solution of an equation similar to the Bethe-Salpeter equation for the masses of mesons, where a quark-quark color pair, or diquark, is one of the quarks. Diquarks, despite the fact that they transfer color, can be real particles in the little-studied color superconducting phase of quark matter [1] (in this phase the color symmetry is broken); therefore, studying the properties of diquarks is also an interesting problem. This work is devoted to the study of the properties of diquarks at a finite temperature and density.

1. Mesons and diquarks in the NJL model

The study of diquarks was carried out within the framework of the Namb-Jona-Lasinio model, the Lagrangian of which for the case with two quark flavors has the form [2]:

$$\mathcal{L}_{\mathcal{NJL}} = \bar{q}(i\partial - \widehat{m_0} - \gamma_0\mu)q + G_s[(\bar{q}q)^2 + (\bar{q}i\gamma_5\vec{\tau}q)^2] \quad (1)$$

G_s– four-quark coupling constant, $\bar{q}$ и q – quark fields, $\widehat{m_0} = diag(m_u^0, m_d^0)$, $m_u^0 = m_d^0$ – quark current masses, $\vec{\tau}$ – Pauli matrices in space $SU(2)$.

The quark mass in the model is given by the gap equation (Schwinger-Dyson equation) (2), which can be obtained by minimizing the scalar pole to a large thermodynamic potential obtained from the Lagrangian of model (1) in the mean field approximation:

$$m = m_0 + 8GN_cN_f mi I_1, \text{где} \tag{2}$$

$$\mathrm{I_1} = -\mathrm{i}\int_{\Lambda}\frac{\mathrm{d^3p}}{(2\pi)^3}\frac{1-\mathrm{f}(\mathrm{E_p}\mu)-\mathrm{f}(\mathrm{E_p}+\mu)}{2\mathrm{E_p}}. \tag{3}$$

Mesons [3] in the NJL model are introduced as collective modes (quark - antiquark bound states). The four-quark interaction in the Lagrangian in the random-phase approximation leads to the appearance of the T-matrix:

$$T_M(k^2) = \frac{2iG_s}{1-2G_s\Pi_M(k^2)}. \tag{4}$$

All information on the properties of a meson is contained in a function called the polarization operator of mesons [4] and which is determined from a polarization loop consisting of quark and antiquark lines (see Fig. 1).

$$\Pi_M(k^2) = i\int\frac{d^4p}{(2\pi)^4}\mathrm{Tr}\,[\Gamma_M S(p+k)\Gamma_M S(p)], \tag{5}$$

где $S^{-1}(p) = (\hat{p} + \gamma_0 p_0 - m)$ - quark propagator, Γ_M - vertex function [5] depending on the type of interaction under consideration. So, for a peony $\Gamma_\pi = i\gamma_5\tau^5$ and the scalar sigma meson $\Gamma_\sigma = 1$.

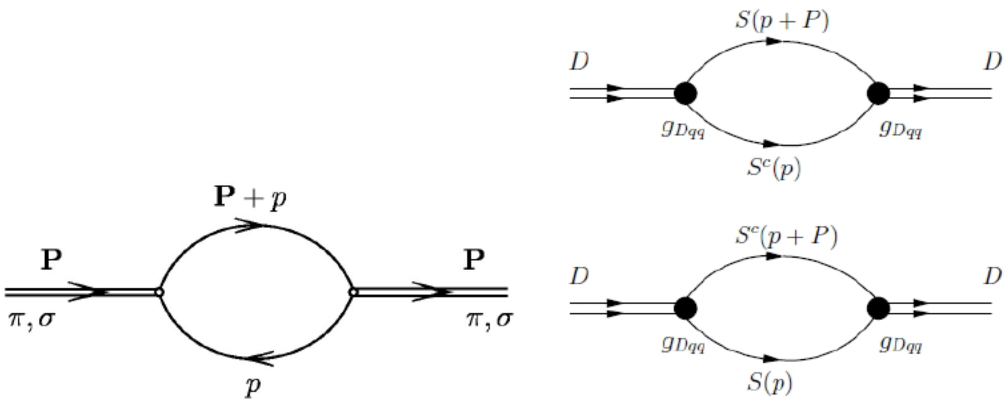


Fig. 1. Meson (left) and diquark (right) loops of the NJL model

To study diquark states, it will be necessary to construct a scattering matrix similar to (2), however, instead of one of the quark lines, one should mean a charge-conjugated quark:

$$\Pi_D(k^2) = i\int\frac{d^4p}{(2\pi)^4}\mathrm{Tr}[\Gamma_M S(p+k)\Gamma_M S^C(p)]. \tag{6}$$

The Bethe-Salpeter equation [6] and polarization operators for calculating the masses of mesons and diquarks are reduced in the NJL model to calculating integrals of the same type, which can be generalized to the case of finite temperatures using the technique proposed by Matsubara [7]. Let's show by the example of mesons:

$$\Pi_{\mathrm{ps}}(\mathrm{k}^2) = \mathrm{N_cN_fI_1} - 2\mathrm{N_cN_f}\mathrm{k}^2\mathrm{I_2}(\mathrm{k}^2) \tag{7}$$

$$\Pi_{\mathrm{s}}(\mathrm{k}^2) = 4\mathrm{N_cN_fI_1} - 2\mathrm{N_cN_f}(\mathrm{k}^2 - 4\mathrm{m}^2)\mathrm{I_2}(\mathrm{k}^2), \tag{8}$$

where the integral I_1 featured in (3), и I_2 has the form:

$$I_2(k^2) = i\int\frac{dp}{(2\pi)^4}\frac{1}{(p^2-m^2)((p-k)^2-m^2)}. \tag{9}$$

2. Calculations

The Nambu-Jona-Lasinio model is non-renormalizable and requires a procedure for regularizing divergent integrals [8]. The most frequently used one is the regularization by cutting off integrals with respect to the three-dimensional momentum. The free parameters of the model (see Table. 1) are fixed from the values of physical quantities: the decay constants of the pion, the density of the quark condensate, and the value of the mass of the pion [9]. The self-consistent solution of the gap equations, the equation for the mass of the bound state, and the constant of weak decay of a pion at zero values of temperature and chemical potential makes [10] it possible to fix the parameters of the NJL model.

Table 1. NJL model parameters

Per.	m_0 [МэВ]	Λ [ГэВ]	G [ГэВ]$^{-2}$	F_π [ГэВ]	M_π [ГэВ]	m [ГэВ]
3D	5.5	0.639	5.227	0.092	0.139	0.31

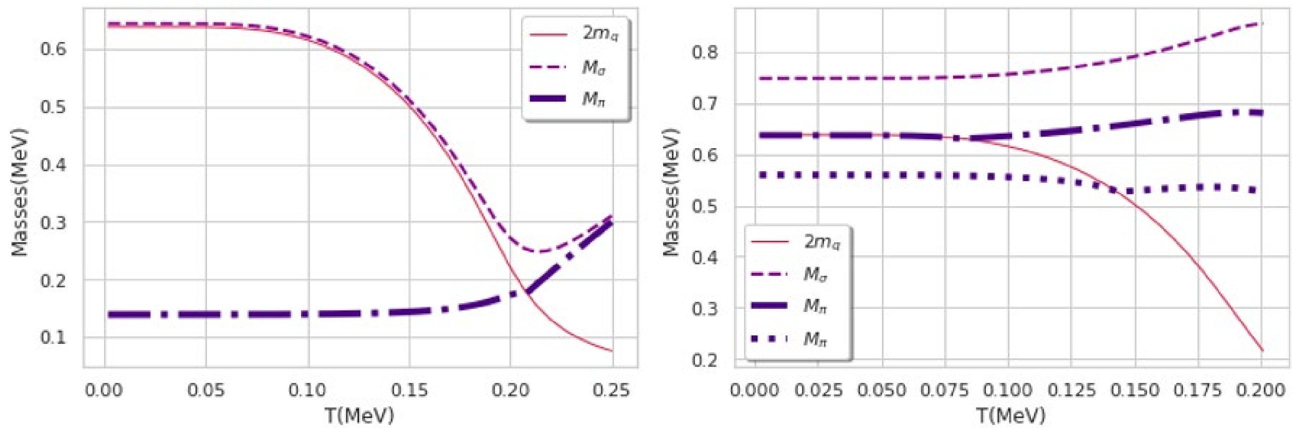


Fig. 2. Masses of scalar and pseudoscalar mesons (left) and scalar diquark (right) as functions of temperature

In Figure 2 shows the temperature dependences of the masses of quarks and mesons in the NJL model. With increasing temperature, the quark mass decreases, while the pion mass increases and, at a certain value of T, coincides with the mass of two quark masses. This temperature is called the Mott temperature and above the Mott temperature, the pion becomes a quasiparticle. In the phase diagram, this transition corresponds to the transition of the hadronic phase of matter into the quark-gluon plasma state. In fig. 2 it is shown that with increasing temperature the mass of the scalar meson decreases and at high temperatures the degeneracy of scalar and pseudoscalar mesons occurs (restoration of chiral symmetry).

The mass of a diquark decreases with increasing temperature and, at a certain value of T, also coincides with the total mass of its constituent quarks. The figure shows that the mass of diquark strongly depends on the coupling constant.

Conclusion

To calculate the diquark mass as a function of temperature, a code written in the FORTRAN language was created for the self-consistent solution of a system of nonlinear integral equations. The generated code is tested on the calculation of the physical characteristics of mesons as a function of temperature within the SU (2) NJL model. A study of the behavior of the mass of a scalar diquark at a finite temperature has been carried out. It is shown that with an increase in the coupling constant, diquarks behave like quasiparticle states. The written code allows one to study the temperature behavior of other types of diquarks.

References

1. Blanquier E. Journal of Physics G: Nuclear and Particle Physics, 2011.
2. Nambu Y., Jona-Lasinio G. Dynamical model of elementary particles based on an analogy with superconductivity I, II. // Physical Review Journals, 1961. – Vol. 122, 124.
3. Klevansky S. P. The Nambu-Jona-Lasinio model of quantum chromodynamics // Reviews of Modern Physics, 1992. – Vol. 64.
4. Kalinovsky Yu. L., Friesen A. V. Properties of mesons and critical points in the Nambu–Jona-Lasinio model with different regularizations // Physics of Particles and Nuclei Letters, 2015. – Vol. 12. – № 6.
5. Parker L., Christensen S. M. MathTensor: a system for doing tensor analysis by computer // Addison-Wesley, 1994.
6. Jones W. T., Fogelin R. J. The Twentieth Century to Quine and Derrida, A History of Western Philosophy // Harcourt Brace College Publishers, 1997.
7. Sheldrick G. M. A Short History of SHELXL // International Union of Crystallography and Oxford University Press, 2006.
8. Arduengo A. J. III, Harlow R. L., Kline M. A stable crystalline carbine // Journal of the American Chemical Society, 1991. – № 113.
9. Booth J. Chatt. The reactions of carbon monoxide and nitric oxide withtertiary phosphine complexes of iron({II}), cobalt({II}), and nickel({II}) // Journal of the Chemical Society, 1962.
10. Hope E., Bennett J., Stuart A. Fluorous zirconium phosphonates: novel inorganic supports for catalysis, in: Pacifichem // International Chemical Congress of Pacific Basin Societies, 2020. – № 961.

THE "CLOUD META SCHEDULER" DEVELOPMENT

Tsegelnik Nikita[1], Balashov Nikita[2]

[1] *Student;*
Dubna State University;
The International School of Information Technologies
"Data Science";
The main educational program:
Physics, group 6161;
e-mail: tsegelnik@jinr.ru.

[2] *Software Engineer;*
Meshcheryakov Laboratory of Information Technology;
Joint Institute for Nuclear Research.

Keywords: cloud technologies, distributed computing, scheduling systems, resource sharing, microservice architecture.

Introduction

The Joint Institute for Nuclear Research (JINR) participates in international scientific projects involving the use of cloud and distributed computing resources. In this regard, the Meshcheryakov Laboratory of Information Technologies (LIT) implemented a cloud infrastructure, which has been actively developing to this day [1].

This infrastructure makes it possible to freely use the services and resources of JINR both for individual users and for various research groups. Each provided service has its own scheduling systems and algorithms that efficiently manage the resources allocated to a specific group of users in the scope of this service. However, there is a problem of rational distribution of resources allocated for all the services and all the user groups, depending on the requirements of a particular group to a particular system, as well as on the real load.

In a particular case, such a task can be solved by an administrator assigned to a certain type of user and the services they use. However, the variety of types of users and services makes it too laborious, and in the long term impossible to implement such an approach. In connection with the above, it has become necessary to develop software dynamically distributing cloud resources between JINR services, depending on the types of users and their requirements for the target systems. In this work, we present our vision of such a software product, a realized prototype, as well as prospects for its further development.

1. JINR cloud resources

The JINR virtual infrastructure management platform is OpenNebula. Further, when we refer to the cloud or cloud resource management, we will mean exactly the OpenNebula platform. All the services that we will consider are distributed and are in some way pre-configured templates, from which virtual machines (VM) are subsequently created and they act as a service with which the user interacts directly or indirectly. Consequently, such systems are both horizontally (number of VMs) and vertically (VM characteristics) scalable. The services of interest include HTCondor, Jupyter, Gitlab (CI Runner).

HTCondor is a high-throughput computing software environment designed for large-scale distributed parallelization of resource-intensive computational tasks [2]. Regular users can send their tasks to the HTCondor queue on the JINR common resources, but for each experimental group there is its own queue on the resources reserved for them. During the processing of experimental data, these resources are actively used, but however, they are often completely idle between processing and the next run of the corresponding experiment, that is, they are not used. Therefore, in the latter case, it seems rational to provide idle resources for other users and/or systems for some time.

The Jupyter project includes software products such as Jupyter Notebook, JupyterHub, and JupyterLab. We plan to implement our service based on JupyterHub, which would allow to automatically create virtual servers in the JINR cloud for processing custom Jupyter Notebooks.

Finally, Gitlab CI Runner is a CI/CD task handler. JINR provides its own Gitlab server, the CI/CD task handler of which is within the scope of our activity.

The aforementioned cloud services are distinguished by the ability to dynamically change their configurations, and, as already mentioned at the beginning of this section, by horizontal and vertical scalability (especially horizontal). Due to the independence of such systems from each other, the management of scaling and reallocation of resources between them must be carried out centrally using third-party software that could take into account the needs of all the cloud services at the same time. We have come up with the Cloud Meta Scheduler as a possible solution, the prototype of which we present in the next section.

2. Prototype implementation

In connection with the above, we can highlight the main features of the final system, which will inherit and reflect our prototype: reliability, fault tolerance, rationality in the allocation of resources, high performance, scalability, logging. Also, for the prototype, it was decided to limit interaction with OpenNebula and HTCondor, excluding division by user type.

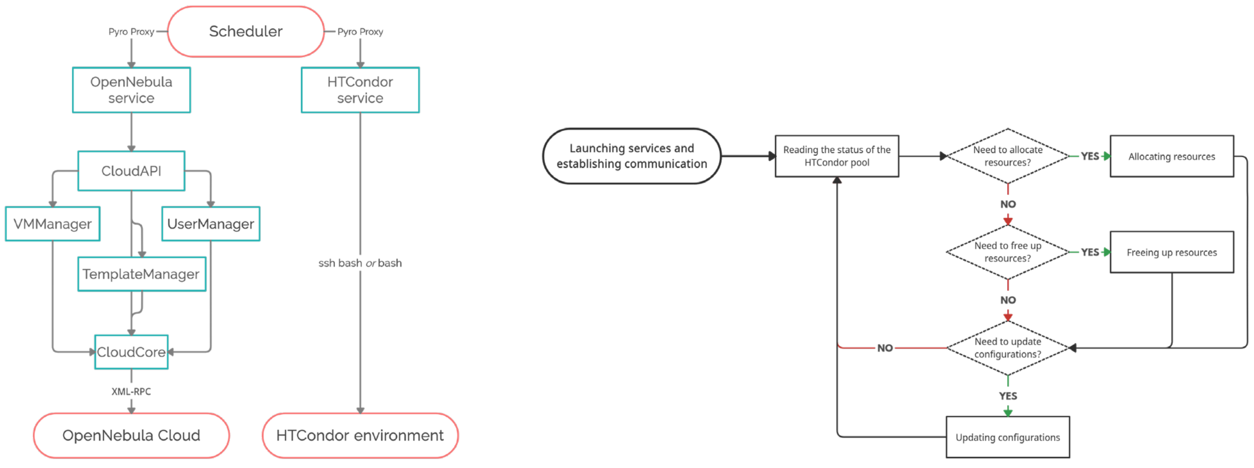
Considering the above requirements, as well as the scale and difference in interaction systems, it was decided to use a microservice architecture [3, 4] and the Python programming language. Now, when we talk about the service, we will mean some microservice implemented by us within the framework of the Cloud Meta Scheduler (CMS). To avoid confusion, we decided to name such services by the same name with the systems for which they will act as an interface. To integrate microservices with each other, we used the Pyro software package [5], which meets the requirements of fault tolerance and high performance.

So, within the framework of the CMS prototype, we have implemented the following three microservices (see Fig. 1a):

1. OpenNebula service, which interacts with the OpenNebula cloud platform via XML-RPC API, for which CloudAPI was implemented; implemented as a Pyro server.
2. HTCondor service that interacts with the HTCondor software environment via CLI or bash scripts; in the case when the interactive node (Submit Node) of the HTCondor environment and the machine on which this microservice is running are different, an ssh connection is used; implemented as a Pyro server.
3. Scheduler service that implements the scheduling algorithm and interaction with all other microservices through a Pyro proxy.

A set of parameters is provided for each service, which is set by the configuration file. There can be several such files (according to each microservice) or one with a common configuration.

The generalized cycle of operation of the entire CMS system is shown in Figure 1b, where the process or action is highlighted by rectangles, and decision making by rhombuses. At the initial stage, before the cycle itself, the configurations are set, all the services are launched, then the Scheduler establishes a connection with OpenNebula and HTCondor, which, in turn, set up a connection (at the stage of their launch) with the OpenNebula cloud and the HTCondor interactive node, respectively. If the connection is successfully established, the status of the HTCondor queue is read, after which a decision is made on the allocation/release of resources. Then, after performing the appropriate action (including the unchanged state), a decision is made whether to update the configuration. Finally, the loop ends with updating the configuration (or skipping this step) and returns to reading the status of the HTCondor queue. It should be noted that if the connection between the Scheduler and the rest of the services is broken, an automatic reconnection occurs, the parameters of which can also be specified in the configuration file.



a) *b)*

Fig. 1. Block diagrams of the implemented Cloud Meta Scheduler prototype: the architecture (a) and generalized system operation cycle (b)

Figure 2 shows a detailed flowchart for the need to allocate cloud resources. Let us briefly explain this scheme. The following parameters are checked and compared with variables: minimum/maximum number of HTCondor worker nodes with the current number of nodes, number of free tasks and free worker nodes, cloud quota and number of involved VMs. Based on the results of the algorithm, a decision is made: whether to allocate resources or not. If there is no need to allocate cloud resources, the cycle moves to checking if some resources need to be released. Otherwise, after resource allocation, the loop proceeds to the configuration update stage.

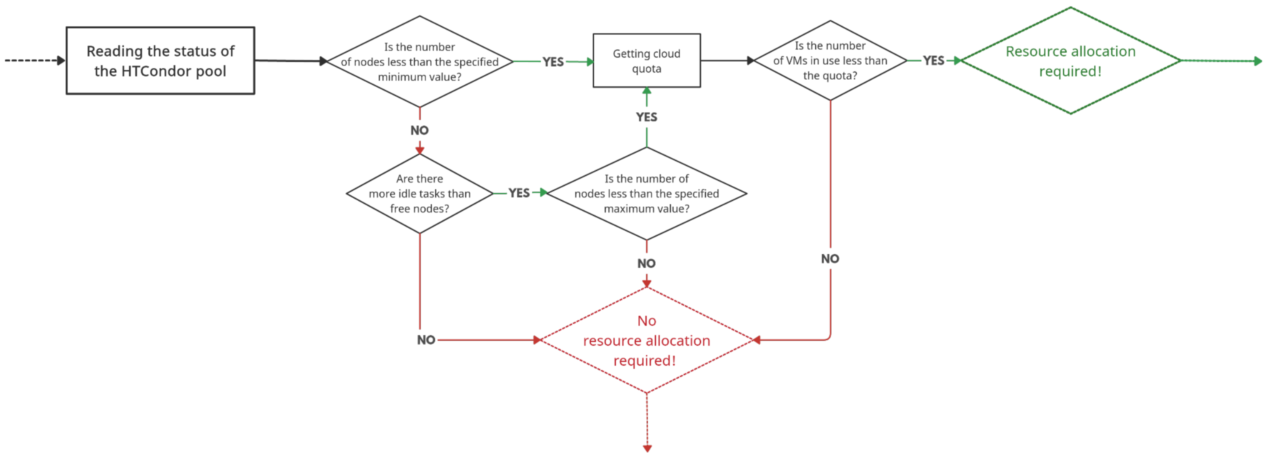


Fig. 2. Flowchart for the need to allocate cloud resources

A flowchart on the need to free cloud resources and update configurations is shown in Figure 3. In the first case, the absence of free tasks is checked, the presence of free nodes, and also whether the number of working nodes is less than the specified minimum value. If the condition is met, then the cloud resources are released, after which the cycle moves on to making the decision to update the configurations. If the condition is not met, then the release stage is skipped. At the decision-making stage, configuration updates are checked for changes in all the configuration files for the CMS services. If there are any, then only the configuration of the service is updated, the file of which was changed during one iteration of the cycle. After this stage, the loop proceeds to the next iteration.

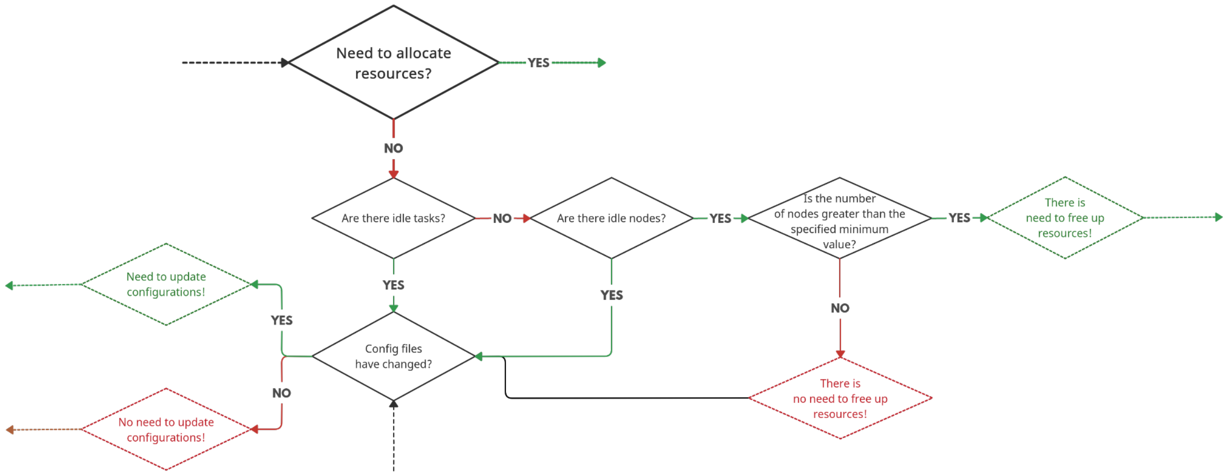


Fig. 3. Flowchart for freeing up cloud resources and updating service configurations

It should be noted that every action of any part of the CMS is fully logged. Superfluous information is logged only in debug mode, which can be enabled by the corresponding configuration parameter.

There are other configuration parameters that we have not listed, for example, the various delays in loop operations or the number of repetitions of these operations if they fail. You can fully familiarize yourself with all the configuration parameters on the project documentation web page [6].

Conclusion

At the moment, the prototype of the Cloud Meta-Scheduler described in the last section, the automatic assembly of the *Docker* image and its automatic update in the JINR registry have been implemented. As already mentioned, the project is documented [6]. The development workflow is organized on the JINR *GitLab* platform. The prototype has been put into test operation, and at the current stage its behavior is being observed and analyzed "in real conditions".

Our future plans include upgrading the prototype based on the results of its test, improving existing services, implementing a more complex scheduling algorithm, implementing automated testing system, introducing new services, and implementing resource sharing between users/experimental groups.

References

1. Baranov A.V., Balashov N.A., Kutovskiy N.A., Semenov R.N. JINR cloud infrastructure evolution // Phys. Part. Nuclei Lett, 2016. – Vol. 13. – № 5(203).
2. Litzkow M., Livny M., Mutka M. Condor – A Hunter of Idle Workstations // Proceedings of the 8th International Conference of Distributed Computing Systems, 1988.
3. Chen R., Li S., Li Z. From Monolith to Microservices: A Dataflow-Driven Approach // 24th Asia-Pacific Software Engineering Conference (APSEC), 2017.
4. Dragoni N. et al. Microservices: yesterday, today, and tomorrow // ArXiv160604036 Cs, 2016.
5. Pyro - Python Remote Objects. – [Electronic resource]. URL: https://pyro5.readthedocs.io.
6. Cloud Meta Scheduler documentation page. – [Electronic resource]. URL: http://cloud-meta-scheduler.pages.jinr.ru/cmscheduler.

Научное издание

Collection of scientific and project activity reports of the graduates of the International School of Information Technologies "Data Science"

Volume 2

В авторской редакции

Подписано в печать 11.08.2021.
Формат 60x84/8. Усл. печ. л. 6.
Тираж 20 экз. Заказ № 60221.

Издательский отдел Объединенного института ядерных исследований
141980, г. Дубна, Московская обл., ул. Жолио-Кюри, 6
E-mail: publish@jinr.ru
www.jinr.ru/publish/