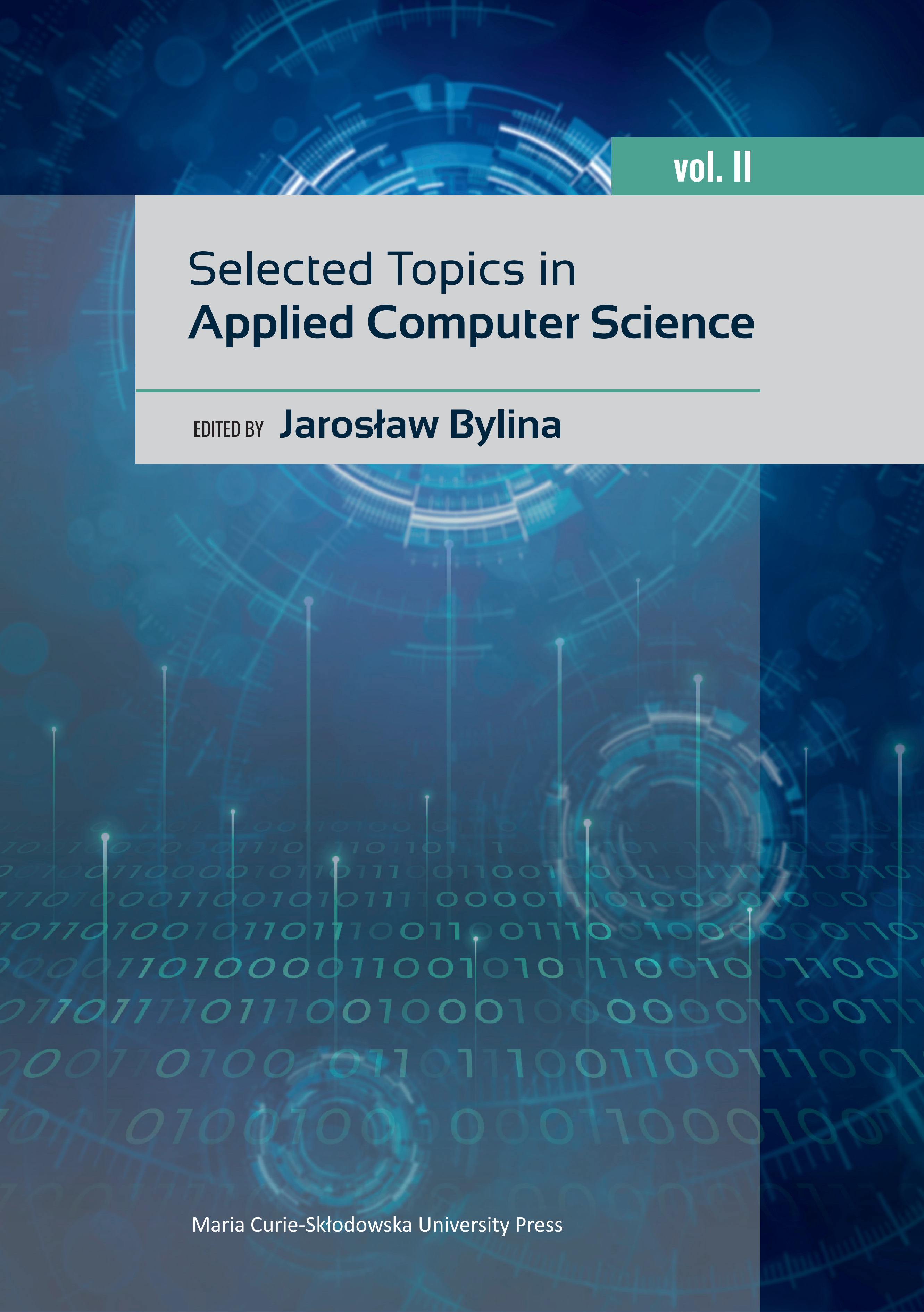


vol. II

Selected Topics in Applied Computer Science

EDITED BY **Jarosław Bylina**

Maria Curie-Skłodowska University Press

Selected Topics in
Applied Computer Science

UMCS

WYDAWNICTWO

vol. II

Selected Topics in Applied Computer Science

EDITED BY **Jarosław Bylina**

Maria Curie-Skłodowska University Press
Lublin 2023

Reviewers
Marcin Blachnik
Adam Domański
Dariusz Mikołajewski
Marek Pałkowski

Cover and front page design
Krzysztof Trojnar

Typesetting
Jarosław Bylina
Grzegorz M. Wójcik

Cover illustration: pixabay.com, freepik.com

ISBN 978-83-227-9675-7

Maria Curie-Skłodowska University Press
ul. Idziego Radziszewskiego 11, 20-031 Lublin, Poland
tel. +48 81 537 53 04
www.wydawnictwo.umcs.eu
e-mail: sekretariat@wydawnictwo.umcs.lublin.pl

Sales Department
tel./fax +48 81 537 53 02
Online bookstore: www.wydawnictwo.umcs.eu
e-mail: wydawnictwo@umcs.eu

Contents

Preface

For several decades, the development of information technology has been gaining momentum. This is reflected in the second volume of the publication prepared by the Institute of Computer Science of Maria Curie-Skłodowska University in Lublin. This time, the book presents works prepared primarily by young researchers — together with their scientific supervisors working at the Institute.

Chapters will raise very current research problems such as:

- the use of Python with GPU for numerical calculations (*Parallel Implementations of the Slope Algorithm in C++/OpenACC and Python/Numba on GPU*);
- an experimental study by data science methods of relaxation techniques influence on measurable body parameters (*Investigating the Influence of Guided Imagery Relaxation on the Selected Electrophysiological Parameters of Human Body*);
- a study of the epidemic and pandemic of the SARS-COV-2 virus — both in terms of social relations as well as the spread of the virus itself — with the use of computer science and mathematical methods (*Network of Social Contacts in Poland of the Times of SARS-CoV-2 Pandemic*, as well as *SARS-CoV-2 Epidemic in Poland and Other Countries*);
- considerations about machine learning used for computer vision — and its future and challenges (*Augmentation is not Enough. On Advancements in Self-supervised Representation Learning for Computer Vision Tasks*).

We hope that this small book will be interesting for everyone who is indifferent to what is happening in contemporary computer science — in particular for students and employees associated with broadly understood IT.

As usual, it is impossible not to thank the many people who were involved in the creation of the book — primarily the authors, but also the reviewers and everyone who contributed to its publication from the editorial and technical side.

Jarosław Bylina
`jaroslaw.bylina@umcs.pl`

Parallel Implementations of the Slope Algorithm in C++/OpenACC and Python/Numba on GPU

Beata Bylina*
Cezary Rumelczyk

1 Introduction

A geographic information system (GIS) is used to process the geospatial data. Therefore, it is essential to accurately estimate topography and topographic attributes. One such attribute is the slope of the terrain serving to generate the slope maps. Slope maps are based on the Digital Terrain Model (DTM), which is most often a raster data structure. Generating slope maps is a computational challenge [7]. A very important issue is the time of generating slope maps, so as not to force the user to wait longer for the result. The parallelization process is used to prevent the occurrence of such cases.

There are several approaches based on parallel computing. One of them is the general-purpose computing on graphics processing units (GPGPU). Nowadays, GPGPU has become very popular. Various programming languages can be used to create parallel code, including C++ and recently the increasingly popular Python.

Modern graphics processors offer opportunities for low-level programmings, such as NVIDIA CUDA (Compute Unified Device Architecture) or OpenCL, and high-level approaches such as OpenACC (Open ACCelerators). The low-level approach aims to explore all the possibilities of the GPU architecture by writing low-level code in C++, thus, it does require considerable effort on the part of the developer. The higher-level programming model greatly simplifies parallel programming on the GPU. When using this model, the programmer needs to indicate the code blocks to be parallelized using the appropriate pragmas.

Another programming language that allows the use of graphics processors is Python. It is faster and easier to program than classic programming languages such as C/C++ and includes many useful libraries for processing geospatial data. However, usability often comes at the expense of performance. It has been argued

*Corresponding author — beata.bylina@umcs.pl that Python applications are considered much slower than C/C++ or FORTRAN applications. Recently, many tools have been developed to counteract these prejudices. One of them is the Numba package. Numba speeds up Python programs on massively parallel NVIDIA GPUs. The efficient implementation of the algorithm for processing geospatial data in Python seems particularly important. Currently, there are few studies in the literature comparing the methods of parallelizing spatial data processing in Python.

In this article, we compare the operation of parallel code written in C++ using the OpenACC standard and in Python using Numba for slope algorithm. The main contributions of this article are as follows:

- Two parallel implementations of slope algorithm on graphics cards, namely C++/OpenACC and Python/Numba.
- Test and run-time evaluation of two parallel implementations of the slope algorithm for different data sizes on cards with two different architectures, namely Kepler and Volta.
- Assessment of the impact of programming languages on the execution time.

The article is structured as follows. Section 2 provides an overview of the related works. Section 3 is dedicated to the theoretical basics of the slope algorithm. Section 4 describes the parallel implementation of the slope algorithm using OpenACC in C++ and Numba in Python. In Section 5, we focus on the details of conducting the numerical tests on GPU and explaining of the results obtained. In Section 6, we present conclusions from the conducted experiment and indicate possible directions for further research.

2 Related works

Much research is focused on designing parallel GIS algorithms and their performance evaluation. This is due to the development of various modern parallel programming models on the GPU, such as CUDA, OpenCL, and OpenACC.

Issues related to parallel programming in C++ using OpenACC have been discussed in works: [5], [9], [10], [12]. The first two papers contain information on comparing parallel programming models on the GPU. The works [9] and [10] concern the use of C ++ and OpenACC in the implementation of algorithms from various fields of science.

In [12], the performance of the CUDA technology is contrasted with OpenACC standard. In terms of kernel uptime, the results showed that OpenACC's performance is lower than CUDA's because the compiler has to translate the OpenACC kernels into object code while the CUDA codes can be run directly. Nevertheless, there are situations in which failure to optimize CUDA-containing code results in longer runtime compared to OpenACC.

Article [5] compares the performance of CUDA, OpenMP, and OpenACC on the Nvidia Tesla V100 GPU under various common scenarios that arise in scientific programming. Evaluation of Tesla V100 memory bandwidth tests showed that the OpenMP and OpenACC compilers are able to generate efficient parallel code.

The work [10] describes high-performance implementations of an example of high-quality recursive pseudorandom number multiple generators, namely MarsLFIB4. Based on the results of his experiments, the author concluded that the obtained portable OpenACC implementation achieves the performance comparable to that of CUDA.

In [9], the watershed analysis algorithm was implemented and evaluated on multi-core central processing units (CPUs) and multi-core graphics processing units (GPUs). The implementations were based on CUDA as well as on the OpenACC. Experimental evaluation showed the advantages of using OpenACC programming over CUDA programming.

Based on the analysis of the results presented in the discussed works [5], [9], [10], [12], OpenACC was selected for the parallel implementation of the slope algorithm on the GPU.

Parallel programming in Python supported by Numba is discussed in the papers: [3], [4], [6]. The articles [3], [4] present using Python language for high-performance computing in science and education. Python performance is considered slow compared to compiled languages such as C, C ++, and FORTRAN. Therefore, in this article, Numba increased the performance of applications written in Python was introduce.

In [6], the author shows the differences between C-CUDA code and CUDA code written in Python with Numba. These differences are presented on the basis of several microbenchmarks analyzed in detail. Additionally, they provide tips on improving Numba applications' performance. Finally, the performance of C-CUDA accelerated Python applications were assessed and compared with the Numba package.

In our work, we evaluate and compare the performance of geoinformatics application, namely the slope algorithm. Additionally, the slope algorithm using the OpenACC standard in C++ and Python with the Numba package is implemented.

3 Slope algorithm

The slope algorithm is the algorithm used to generate slope maps from digital elevation models (DEM). There are several algorithms for determining the slope, differing from one another by the calculation method [2], [11]. All of these methods use a moving mask with a size of three by three cells as shown in the Table 1.

Table 1: Mask used in the neighborhood method to calculate the slope in cell number 9

z_1	z_2	z_3
z_8	z_9	z_4
z_7	z_6	z_5

There are several numerical methods for calculating the value of the slope. The slope algorithms differs in the way of calculations due to the number of neighbors (2, 3, 4, 8, 9). Later in the paper we will consider only one neighborhood method called the Horn algorithm (as [2]). The Horn method counts among the 8 neighbors.

This method is used in the ArcGIS [13] and QGIS [18] desktop software. The slope value of the middle cell of this mask with the number 9 is calculated.

The formula for this method in radians is as follows:

$$S = \operatorname{atan}\left(\sqrt{(S_{e-w}^2 + S_{n-s}^2)}\right) \tag{1}$$

S_{e-w} denoted east-west slope and is described by the formula:

$$S_{e-w} = \frac{(z_3 + 2z_4 + z_5) - (z_1 + 2z_8 + z_7)}{8d} \tag{2}$$

S_{n-s} denoted north-south slope and is described by the formula:

$$S_{n-s} = \frac{(z_1 + 2z_2 + z_3) - (z_7 + 2z_6 + z_5)}{8d} \tag{3}$$

where:
S — slope angle in radians
from z_1 to z_8 — elevation of cells 1 through 8
d — cell resolution

This method does not consider elevations at the center of a nine-element mask. It leads to inaccurate slope estimates if the elevations in question have small pits or peaks. However, despite the loss of local variability in the output slope map, this method can be useful if the input DEM is inaccurate [2].

4 Parallel implementations

This section provides basic information about the technologies used for creating two parallel implementations on GPU. One of the implementations written C++ will use the OpenACC standard. The second one, coded in Python, will use the Numba library.

4.1 GDAL

GDAL (The Geospatial Data Abstraction Library) [14] is a computer software library for reading and writing raster and vector geospatial data formats. This library presents a single abstract data model for the calling application for all supported formats. Functions from the GDAL library will be used to read the file with raster data in the TIFF format and to save the file with the slope created. This part of the code will not be written in parallel.

To process raster data, they must be stored in operating memory. Because the data are in the form of a raster — that is a matrix, they are often stored as a two-dimensional array. Another way of representing them in the random access memory is to store the matrix:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

(where m is the number of rows and n is the number of columns) in a one-dimensional array. According to one of the storage schemes for matrices of general form, presented in the LAPACK [1] library and the convention for storing two-dimensional arrays in C/C++ languages, the matrix elements are stored row-by-row (that is, in *row-major order*):

$$A = \big(a_{11}, \dots, a_{1n}, a_{21}, \dots, a_{2n}, \dots, a_{m1}, \dots, a_{mn}\big).$$

4.2 OpenACC

OpenACC [17] is a parallel computing programming standard developed by Cray, CAPS, Nvidia, and PGI. The standard is designed to simplify parallel programming of heterogeneous CPU-GPU systems. OpenACC supports *C*, *C++* and *Fortran*. In the case of this work, the version written in C++ will be used. It is worth mentioning that the above library is portable and runs on various hardware architectures. OpenACC is based on pragmas and offers a set of directives for parallel code execution on the GPU or CPU. The use of OpenACC directives allowed us to parallelize our code without having to explicitly change the sequence code. In particular, to parallelize the slop calculations using the formula (1), parallelism at the loop level was used along with the corresponding clauses for optimization. We used the PGI compiler to compile our code. The code snippet is shown in Listing 1.

4.3 Numba

Python is a programming language used in industry and academia. One reason is that it is easier to learn than classic programming languages like C/C++. However, Python's performance is considered slow compared with compiled languages such as C, C++, and FORTRAN. A collection of libraries such as NumPy and Matplotlib or Scipy provides a rich set of functions for scientific computing in Python.

Numba [15] is a technology that translates Python functions into machine code (parallel to processor or graphics accelerator) at runtime, using the standard LLVM (Low Level Virtual Machine) compiler library. Python algorithms compiled by Numba can come close to the speed of C++ or any other compiled language.

NumPy [16] is the essential Python scientific computing package. It is a Python library that provides a multidimensional array object, various derived objects, as well as a range of routines for quick array operations including math, logic, shape manipulation, sort, select, input/output, and more.

Numba is designed for use with NumPy arrays and functions. Numba generates specialized code for various array data types and layouts to optimize performance. Numba was used to optimize the parallel implementation of the slope algorithm on GPU in Python language. The code snippet is shown in Listing 2 and the code snippet in parallel Python — in Listing 3.

```
tab[0:n] = dem[nRows][nCols]; //?
#pragma acc data copy(tab [0:n], slope [0:n])
{
  #pragma acc parallel
  {
    #pragma acc loop collapse(2)
    for (r = 1; r < nRows - 1; r++)
    {
      for (c = 1; c < nCols - 1; c++)
      {
        z1 = (r - 1) * nCols + (c + 1);
        z2 = (r * nCols) + (c + 1);
        z3 = (r + 1) * nCols + (c + 1);
        z4 = (r - 1) * nCols + (c - 1);
        z5 = r * nCols + (c - 1);
        z6 = (r + 1) * nCols + (c - 1);
        z7 = (r + 1) * nCols + c;
        z8 = (r - 1) * nCols + c;

        p = ((tab[z1] + 2 * tab[z2] + tab[z3]) -
            (tab[z4] + 2 * tab[z5] + tab[z6]))/8;

        q = ((tab[z6] + 2 * tab[z7] + tab[z3]) -
            (tab[z4] + 2 * tab[z8] + tab[z1]))/8;

       slope[r * nCols + c] = atan(sqrt((p * p) + (q * q)));
      }
   ...
}
```

Listing 1: C++/OpenACC implementation

```
@jit
def Slope(rows, cols, dem, slope):
   for r in prange(1,rows-2):
      for c in prange(1, cols-2):

         p = ((dem[r-1][c+1] + 2*dem[r][c+1] + dem[r+1][c+1]) -
             (dem[r-1][c-1] + 2*dem[r][c-1] + dem[r+1][c-1]))/8

         q = ((dem[r+1][c+1] + 2*dem[r+1][c] + dem[r+1][c-1] -
             (dem[r-1][c+1] + 2*dem[r-1][c] + dem[r-1][c-1]))/8

         slope[r][c] = np.arctan(np.sqrt(p*p+q*q))
```

Listing 2: Python/Numba implementation

```
@cuda.jit
def Slope(dem, slope):
    r,c=cuda.grid(2)
     if r>0 and r<dem.shape[0]-1 and c>0 and c<dem.shape[1]-1:
     p = ((dem[r+1,c-1] + 2*dem[r+1,c] + dem[r+1,c+1]) -
         (dem[r-1,c-1] + 2*dem[r-1,c] + dem[r-1,c+1]))/8

     q = ((dem[r+1,c+1] + 2*dem[r,c+1] + dem[r-1,c+1]) -
         (dem[r+1,c-1] + 2*dem[r,c-1] + dem[r-1,c-1]))/8

     slope[r,c] = math.atan(math.sqrt(p*p+q*q))
```

Listing 3: Parallel Python/Numba implementation

5 Numerical experiment

This section describes details of the numerical experiment with the results and their interpretation.

5.1 Methodology

We benchmark two versions of the parallel slope algorithm (both using the single precision floating-point number format), namely:

- `C++OpenACC` using OpenACC Standard in C++ language.
- `Python-Numba` using Numba in Python language.

Four different DEMs sizes were considered to evaluate the execution time, the speedup and the scalability of implementations with the data size increasing. The sizes of data used for testing are presented in Table 2. These DEMs represent the models of digital terrain in Southeast Europe downloaded from the Land Monitoring Service (https://land.copernicus.eu).

To determine the execution time of two parallel slope implementations, tests were carried out on two systems with GPU architecture, namely: K40 representing the Kepler and V100 representing the Volta generation. This made it possible to assess whether the same or different behaviors could be observed in different architectures. Each test was executed 10 times. Table 3 shows the details of the

Table 2: Test area data details

Name	No. of columns =No. of rows	No. of elements	size of array in GB
5 × 5	5 000	25 000 000	0.1
10 × 10	10 000	100 000 000	0.4
20 × 20	20 000	400 000 000	1.6
40 × 40	40 000	1 600 000 000	6.4

Table 3: Hardware specifications of individual compute nodes

KEPLER	CPU	2x Xeon E5-2670 v3 2.30GHz
	RAM	128GB
	GPU	NVIDIA Tesla K40m
VOLTA	CPU	2x Xeon E5-2670 v3 2.30GHz
	RAM	128GB
	GPU	NVIDIA Tesla V100s

hardware specifications where both test platforms use *Intel Xeon Processor E5-2670 v3* processors. Each processor has 12 cores. The machine is equipped with 2 such processors with 12 cores each, i.e. included 24 physical cores. KEPLER has a Tesla K40m graphics card, which is characterized by a 745 MHz core clock, 12 GB memory and 2880 cuda cores. VOLTA has a Tesla V100s graphics card, which is characterized by a 1370 MHz core clock, 32 GB memory and 5120 cuda cores.

The following software is installed on each machine:

- operating system: CentOS 7.5
- kernel: Linux 3.10.0
- GDAL: 3.1.4
- pgi: 20.4
- Numba: 0.53.1

The pgi compiler is used with the optimization flags: `-fast` and `-acc` for `C++OpenACC`.

5.2 Comparing the execution times and the speedup

This section presents the final results of the execution times. In Table 4 the execution times for both `C++OpenACC` and `Python-Numba` for four different DEMs size on two GPU and sequential program execution on single-core CPU are shown.

Table 4: Times of the execution in seconds for two versions of the implementation for four different DEMs size on different machines

Versions	Machines	5 × 5	10 × 10	20 × 20	40 × 40
`C++OpenACC`	KEPLER	0.80	**1.08**	**2.14**	-
	VOLTA	1.21	1.40	2.39	6.39
	SEK CPU	**0.73**	2.5	9.95	33.09
`Python-Numba`	KEPLER	3.29	3.76	5.68	-
	VOLTA	1.89	2.10	2.79	**5.04**
	SEK CPU	797.57	2820.85	12227.42	38583.86

According to these experiments, CPU and GPUs execution times increase with DEM size, as expected. Sequential program execution (SEK CPU) in C++ is more efficient than sequential program execution in Python, which is commonly known [3], [4], [6]. In the first data set, single-core implementation on single-core in C++ (`C++OpenACC`) outperforms GPU implementations about 10% on KEPLER and about 66% on VOLTA. This is due to two facts. Firstly, C++ compiler produces code that executes sequentially very quickly. Secondly, the first data set takes up very little space in the memory of both cards (Volta and Kepler), which is associated with a relatively small amount of calculations. Implementations on GPU always outperform their counterparts on single-core for other sizes.

Implementations in Python (`Python-Numba`) on GPU always significantly outperform the sequential implementation on a single-core. `C++OpenACC` applications outperform `Python-Numba` versions for all data except the largest DEM data set comparing CPU and GPU, respectively. The largest DEM data set does not fit on the Kepler card because the card has only 12 GB of memory and the largest data set is as large as 6.4 GB. Each implementation works on two DEMs, which means it takes 12.8 GB of memory. The results showed that in the case of the tested `Python-Numba` application, for large data sizes on the Volta card, the execution time was faster about 27% than the execution time of the `C++OpenACC` implementation.

For better performance evaluation, we calculated the speedup to compare the execution time of parallel implementations on GPU of slope algorithm with the execution time of sequential on a single core CPU processor. This dependence is illustrated in Table 5.

Table 5: The Speedup

Versions	Machines	5×5	10×10	20×20	40×40
`C++OpenACC`	KEPLER	0.91	2.31	4.65	-
	VOLTA	0.60	1.79	4.16	5.18
`Python-Numba`	KEPLER	242.42	750.23	2152.71	-
	VOLTA	421.99	1343.26	4382.59	7655.53

The speedup of the `Python-Numba` outperforms the speedup of the `C++OpenACC`. The results showed also practical differences between KEPLER and VOLTA for both implementations.

6 Conclusion

In this article, we introduce a methodology that allows you to speed up raster computation for the slope algorithm on the GPU using two different programming languages, namely C++ and Python. Different parallelization technologies were used for each language. Namely, OpenACC was used for C++ and Numby was used for Python. We concluded that in C++ it is worth using OpenACC, which greatly simplifies the programming process, and the resulting code is simple, readable, and efficient at the same time.

Based on the tests performed, we can also see that both implementations scale well with increasing problem size. The only limitation connected with the size is the size of the DEM, which must fit in the memory of the graphics card. In Python, the use of Numby produces very efficient parallel code that can compete with a C++ implementation in terms of performance.

The proposed parallelization methods can be extended to related algorithms, such as edge detection in digital image processing. This operation removes information such as color and brightness and leaves only the edges. For edge detection algorithms, the cross operator Sobel [8] will be considered. The approach too may be useful for creating high-performance, scalable code suitable for GPU-based computing systems for processing geospatial data. In addition, we plan to further improve the performance of the slope algorithm. We want to use Unified Memory (UM) which proposes program abstraction of uniform memory space on CPU and GPU for Python implementation.

References

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney,*LAPACK Users' Guide (Third)*, Society for Industrial and Applied Mathematics, 1999.

[2] M. Dunn, R. Hickey, *The Effect of Slope Algorithms on Slope Estimates within a GIS*, Cartography, v. 27, no. 1, p. 9–15, 1998.

[3] A. Marowka, *On parallel software engineering education using python*, Educ Inf Technol 23, p. 357–372, 2018.

[4] A. Marowka, *Python accelerators for high-performance computing*, J Supercomput 74, p. 1449–1460, 2018.

[5] K. Mikhail, A. Timoveev *Performance analysis of CUDA, OpenACC and OpenMP programming models on TESLA V100 GPU*, J. Phys.: Conf. Ser. 1740 012056, 2021.

[6] L. Oden, *Lessons learned from comparing C-CUDA and Python-Numba for GPU-Computing*, 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), pp. 216–223, 2020.

[7] H. Qunying, Y. Chaowei, *Optimizing grid computing configuration and scheduling for geospatial analysis: An example with interpolating DEM.*, Computers and Geosciences, v. 37, p. 165–176, 2011.

[8] I. Sobel, G. Feldman, *An isotropic 3x3 image gradient operator*, Stanford Artificial Intelligence Project (SAIL), 1968.

[9] N. Stojanovic, D. Stojanovic, *Parallelizing Multiple Flow Accumulation Algorithm using CUDA and OpenACC*, ISPRS Int. J. Geo-Inf. 8, 386, 2019.

[10] P. Stpiczyński, *Algorithmic and language-based optimization of Marsa-LFIB4 pseudorandom number generator using OpenMP, OpenACC and CUDA*, Journal of Parallel and Distributed Computing, Volume 137, p. 238–245, 2020.

[11] J. Tang, P. Pilesjö, *Estimating m from raster data: a test of eight different algorithms in flat, undulating and steep terrain*, WIT Transactions on Ecology and the Environment, v. 146, p. 143–154, 2011.

[12] L. Xuechao, S. Po-Chou, *An Early Performance Comparison of CUDA and OpenACC*, MATEC Web of Conferences 208(1):05002, 2018.

[13] ArcGIS, https://desktop.arcgis.com/

[14] GDAL, https://gdal.org/

[15] NUMBA, https://numba.pydata.org/

[16] NumPY, https://numpy.org/

[17] OpenACC, https://www.openacc.org/

[18] QGIS, https://docs.qgis.org/

Investigating the Influence of Guided Imagery Relaxation on the Selected Electrophysiological Parameters of Human Body

Katarzyna Zemła[*]
Grzegorz M. Wojcik
Filip Postępski
Łukasz Kwaśniewicz
Andrzej Kawiak

1 Research background and the existing state of knowledge

Hypnosis, hypnotherapy and techniques like Guided Imagery (GI) are widely recognised as method supporting a wide range of therapies, including oncotherapies and mental disorders.

The primary objective of this paper is to present the literature review of the relaxation techniques appliance in supporting the health recovery programs is presented.

The secondary objective of this paper is to conduct the pilot study aimed at measurement of electrophysiological parameters: EEG brain cortical activity, pulse and blood saturation of the patient exposed to Guided Imagery hypnosis.

There are numerous examples of using hypnotherapy in the treatment of patients affected by HIV, ARC, or AIDS, among others [21]. For example, Auerbach demonstrated a meaningful reduction in physical symptoms associated with HIV, such as fever, pain, nausea, and a significant increase in activity and resilience in case of patients with ARC and AIDS who participated for 8 weeks in a group program that used biofeedback, imagery, and hypnosis, as compared to a control group [1]. Gochros used hypnosis in simultaneous individual and group therapy of seropositive patients in order to strengthen their ability to cope with the diagnosis and reduce the resulting stress [21]. His results showed a positive effect of hypnosis

[*] Corresponding author — kzemla1@st.swps.edu.pl

on anxiety and helplessness. Mentioned 8-week-long group program of Kelly, which included self-hypnosis and meditation training, was shown to help reduce stress and improve self-control and the daily quality of life of patients [14].

Newton and Marx used imaginal hypnosis in the Simonton approach with 4 men (10 individual sessions) and 22 men (10 group sessions) in order to improve the long-term survival of the patients [19]. Significant reduction of stress decreased anxiety related to their condition, and increased activity was observed in the case of patients who received the individual therapy.

The abovementioned Simonton method was first used in 1971 by Carl Simonton, an American physician and radiation oncologist. It had been then developed for more than 30 years. Simonton introduced the systematic use of psychotherapeutic interventions as a necessary extension of conventional cancer treatment [9, 29, 30]. Criticism of his studies and his reports on the positive therapeutic results of his approach to the treatment of patients affected by cancers initiated long series of standardized clinical trials. For instance, David Spiegel [33] confirmed the effectiveness of this approach.

Patients with distant metastases of advanced breast cancer were divided into two groups. Patients who additionally participated in a cognitive-behavioral therapy program as an adjunct to standard cancer therapy showed significantly better outcomes than patients in the control group who were only treated according to the current standards. Fawzy [8] came to similar conclusions regarding psychotherapeutic intervention in the treatment of patients diagnosed with malignant melanoma. And despite the clinically proven beneficial results of the use of cognitive-behavioral therapy increasing the level of coping with the situation after diagnosis, reducing the stress experienced, and having a beneficial effect on life expectancy after diagnosis has not resulted so far to attach such a standard of treatment support to all patients although it is known that the lifespan of the included participants in Fawzy's study was statistically twice as long [8]. The innovative concept to help patients using VR methods has the potential to change that and enable patients to support their treatment from the psychological edge. It is known and proven that when patients "think healthy" it supports recovery because they are able to:

- Enter a state of relaxation and relaxation as often as possible. Before and after, but also, if possible, during medical procedures.

- Put the brain into an alpha state and imagine positive scenarios of how my body, organs, and cells are healing under the influence of the applied therapy.

- Understand that these visualizations and alpha state are the way to support the immune system, as well as a pathway in its conditioning process.

- On the grounds of experiences (including those from virtual reality) build realistic and positive beliefs about one's condition, the medical procedures used, and the processes of treatment and recovery.

From the neuroscientific perspective adopted by Rossi [24], it is the patient's creative activity that generates, through the neuroplasticity of the brain, new neuronal connections so-called "miracle of healing based on the body-mind relationship." This deeply meaningful, internal creative mental process produces a hypnotic experience for problem-solving problems and healing. Healing is located within the patient. The therapist has no secret powers to control or heal. Patients heal themselves if they are lucky enough to receive the right "therapeutic suggestions" and psychological support which is described as "implicit heuristics of processing."

Research on implicit processing heuristics should take advantage of the current level of neuroscience and computer data processing and available technologies to build upon that [7]. For this purpose, our EEG study allows, among other things, the analysis of the amplitude and frequency of brain waves under hypnosis.

Several basic waves can naturally appear in the EEG recordings:

- Alpha waves (frequency 8–13 Hz, amplitude 30–100 μV) — are the rhythmic activity of the cerebral cortex in the 8–12 Hz range. This is one of the earliest observed structures (graphemes) of the EEG. The occurrence of the frequency of the rhythm alpha is attributed to the state of relaxation with eyes closed. Alpha waves are best seen in the posterior (occipital) leads, that is, from around the part of the cortex responsible for processing visual information. The alpha rhythm is of fundamental importance in EEG analysis of sleep. Although it does not occur during actual sleep it is indicative of the patient's "pre-sleep" wakefulness, and its disappearance signifies the transition from the waking state to shallow sleep. They are also attributed to a state of rest. Reduced alpha wave amplitude is noted in stressed individuals and those with an elevated state of anxiety.[1].

- Beta waves (frequency 12–30 Hz, amplitude >30 μV) In the beta spectrum, the following compartments are distinguished: slow beta waves (12–15 Hz), the proper intermediate beta band (15–18 Hz), and fast beta waves, with frequencies above 19 Hz. This unsynchronized neuronal activity characterizes the usual daily activity of the cerebral cortex in humans. The range of this frequency is observed during the state of active functioning, wakefulness, and alertness. It increases during logical thinking when attention is directed to cognitive tasks and the external world.[2]

- Theta waves (frequency 3.5–8 Hz) — activity in the frequency band from 3 to 7 Hz and a spread of several tens of μV. Characteristic theta waves occur, for instance, during the period of shallow sleep — it is assumed that during this time the assimilation and consolidation of learned content take place. Theta

[1] Low alpha (8–10 Hz) — is the range of waves with a frequency below the peak of alpha in the test person, with the eyes closed. With age, a decrease in the peak frequency of this wave. The higher peak frequency of this wave is found in more cognitively fit individuals. This frequency band is associated with meditation, with maintained calmness and relaxation. Low alpha is subject to diurnal fluctuations and we can note its higher amplitudes between the hours of 11 a.m. to 3 p.m. Significant fatigue of the subject can also affect the spectrum of this waveform [38]. High alpha (11–12 Hz or 11–13 Hz) — this frequency occurs when the state of high awareness of the environment. In this state, the brain can react quickly a precisely to changes in the environment. Waves of this band are a state of mental and physical calm, also known as the "zona" state. The mind is focused on the given moment "here and now," It is a state associated with high concentration and certainty of action [11].

[2] SMR sensory rhythm (13–15 Hz) — is observed in the sensory band of the cortex cerebral cortex. It is a spindle-shaped waveform. It determines the state of alertness, but without muscle tension muscles. It is a state in which high concentration is achieved. An understated amplitude of this wave may indicate problems with maintaining focused attention [11].

waves are the most common present brain waves during meditation, trance, hypnosis, intense dreaming, and intense emotions. It is mainly observed In the medial part of the front part of the cerebrum.[3]

- Delta waves (frequency 1–3 Hz) are high-amplitude activity with a low frequency (0–4 Hz) and a duration of at least 1/4 s. For practical purposes, the lower limit of frequency was assumed to be 0.5 Hz. Appearing during deep sleep, delta waves with an amplitude of more than 75 μV are called slow waves (SWA). Their appearance is due to the high synchronization of cortical neurons (a higher one is encountered only during an epileptic attack). Delta waves are also recorded during deep meditation in young children and the case of certain types of brain damage.[4]

- Gamma waves (frequency 25–100 Hz) — activity in the Hz frequency band is referred to instead referred to as high-frequency (high) gamma. The gamma rhythm accompanies motor activity and motor functions. Gamma waves are also associated with higher cognitive processes, including sensory perception, and memory, among others. It is speculated that gamma rhythms modulate perception and consciousness and that the greater appearance of gamma waves relates to expanded consciousness and spirituality [11].

Regardless of culture, race, upbringing, religion, and political views, all people with biologically intact brains experience stress in the same stereotypical way. The basis of such a stereotypical response to life-threatening situations are neurophysiological processes, related to the stimulation of the relevant areas of the central nervous system, which influences the immune system through the autonomic nervous system (sympathetic and parasympathetic), the endocrine system, and a direct effect on the limbic-hypothalamic system secreting immunomodulating neuropeptides [43]. This allows one to measure how even stagnant stress levels may change when applying stress reduction factors such as relaxation and visualization. Knowing that study conducted in 1987 by Kempthorne-Rawson, Persky, and Shekelle proves that pessimism and depression contribute to higher mortality among patients with cancer such methods to reduce its level should be included in standard treatment. In the 1950s, West, Blumberg, and Ellis showed that the rate of tumor growth is more related to psychological factors than to the degree of tumor differentiation found on histopathological examination [42].

Knowing how the body behaves in a relaxed or stressed state, it is possible to construct tests and use such measurements that will collect signals from heart, skin,

[3] Theta waves are associated with the extraction of information from memory and the ability to control reactions to stimuli. At this frequency, we are aware of our surroundings while the body is in a state of deep relaxation. They are associated with conscious observation of the environment (thalamic nuclei of the brain). In the state of theta waves, very creative thoughts, inspirations, and imaginations. This frequency helps recall memories, fantasies, and associations. In contrast, excessive amounts of theta waves have been reported in people with attention deficit disorder [11].

[4] Delta waves are the slowest of all brain waves. They occur during deep sleep and account for more than 50% of recorded brain activity. They have also been observed during transcendental meditation. Information received at this level is usually unavailable at the level of consciousness. Delta waves dominate the QEEG spectrum in infants up to 6 months of age. age. They are also recorded in brain damage and in brain tumor diagnoses [11].

or brain activity, so that researchers will be able to prove that hypnosis brought expected changes within the patient's body. For example, using measurements such as respiratory rate per minute, duration of inspiration and expiration, tidal volume (in ml), heart rate HR (in beats/minute), respiratory sinus arrhythmia RSA (difference between the maximum and minimum heartbeat interval, in ms,) logarithm of HFHRV-transformed power in the high-frequency band of heart rate variability, LF-HRV-transformed power in the low-frequency band of heart rate variability can quantitatively demonstrate how appropriately timed relaxation by modifying breathing patterns can put subjects into a relaxed state [39]. It is often assumed that cardiorespiratory changes induced by breathing instructions trigger a relaxation response [5]. Psychologically, breathing techniques usually induce an increased focus on internal sensations and comparatively disregard external stimuli [40]. Physiologically, most breathing exercises are designed to decrease sympathetic activity and increase the parasympathetic activity of the nervous system [2]. Results from a study at the University of Leuven strongly suggest that voluntary changes in the length of inhalation in comparison to exhalation are an important determinant of participants' reported relaxed states [39].

EEG studies on relaxation, on the other hand, show that a decrease in total power in the entire cerebral cortex during the relaxation state means that the brain activity of individuals during the relaxation process gradually decreases [36]. Physiological indicators of responses to relaxation introduced by Foster [17] include reductions in oxygen consumption, respiration and heart rate, as well as an increase in the production of alpha brain waves. Increased power of alpha and theta frequencies and interhemispheric synchronization, especially frontal alpha coherence [37] are usually considered as neurophysiological indicators of sensorimotor state and mental rest.

Regular relaxation practice can affect various physiological and psychological parameters related to aging, digestion, general well-being, and psychosomatic diseases. Consequently, there is a growing need to monitor physiological processes related to relaxation and stress response [25]. From the current literature on the subject, it can be concluded that deep relaxation is most often led to by slow, deep breathing at a frequency of 0.1 Hz.

In an article [25], the authors confirm that 6 breaths per min promote relaxation. In a book entitled "Relaxation, Meditations & Mindfulness" [31] mentions techniques to achieve a state of deep relaxation. These include Yoga classes, where progressive muscle relaxation and deep breathing occur. The author points out that the breath should be slow and even, and sometimes deep or shallow. Relaxation breathing has a rhythm in which the exhalation is slow and steady. At first, it may be deep and later shallow without effort. In general, relaxed exhalation takes twice as long (6 seconds) as inhalation (3 seconds).

We can divide the breathing process itself into:

- normal breathing (eupnoe) with a frequency of 0.25 Hz or 25 breaths per min,
- slow breathing (bradypnoe) with a frequency of 0.1 Hz or 6 breaths per min,
- fast breathing (tachypnoe) with a frequency of 0.5 Hz.

In the paper [4], researchers examined the effect of the respiratory cycle on EEG. W order to do so, they compared the spectral analysis of the EEG signal during inspiration and expiration.

Normal, slow, and fast breathing were checked. The researchers noted that during inhalation with normal breathing, delta wave activity in the parietal region and total activity in the frontal region. With fast breathing during inspiration, there is a decrease in beta wave activity in the central region and activity in the theta in the posterior temporal and occipital regions. Compared with the EEG in eupnoe, bradypnoe and tachypnoe, there was a decrease in the spectral power of all spectral bands except delta during faster respiration rates and vice versa, with a significant difference found mainly between bradypnoe and tachypnoe, less frequently between eupnoe and tachypnoe.

In another article [10], researchers examined the effect of breathing patterns on EEG activity. They conducted the study on healthy participants. Each examined had to breathe deeply and slowly (6 breaths per min), hold their breath, and breathe quickly and deeply (30 breaths per min). The EEG signal was read from the frontal, parietal and occipital regions of the head. The researchers detected an increase in alpha and beta activity in the frontal region during deep and slow breathing. In contrast, there was a process of decrease in the activity of these waves in all regions during breath-holding. In the case of slow and deep breathing, only alpha decreased.

The pace of speech we know from studies on the subject is that a healthy person utters about 10–15 sounds per second. In the case of uttering a greater number of them, i.e. 20 (or more), understanding the speaker's speech is much more difficult.

Three modes of speaking tempo can be distinguished [34]:

- lento (slow, slow tempo),
- moderato (moderate),
- allegro (fast, English quick tempo).

Usually, texts are spoken at a moderato tempo. For longer speeches, as a rule, there are different speaking tempos. Their interplay is a characteristic feature of spoken language. a person pronounces an average of 10–15 vowels per second. The pronunciation of 21 vowels per second is on the verge of speech intelligibility. The time taken to pronounce syllables and vowels is measured in milliseconds. For example, the duration of shortness consonants is about 40 milliseconds, while the duration of an average syllable is 200–300 milliseconds. In fast speech, the average duration of a vowel is 60–70 milliseconds, and in slow speech — 150–200 milliseconds. If only in connection with this knowledge, it would be necessary to adjust the recording in such a way that the speaker speaking to the patient pronounces the voices at a rate that is within the 150–200 millisecond range. Science is studying also the effect of sound on our mood. Human responses to sound are experienced on several different levels: physical, mental, cognitive, and behavioral [38].

Nevertheless, there are still relatively few studies that document the relevance of this factor, especially at the level of interpersonal communication. Instead, we know that one of the elements of effective psychotherapy is empathy, which also expresses itself in non-verbal ways [17]. That's why it's so important to lean into the importance of speech characteristics, to consciously use the right sounds, tone of voice, or tempo, as this translates into the reactions of physiological and mental reactions that are induced in the recipient especially by using VR tools.

In the professional exchange of information, for example, in the psychotherapeutic process, communication takes place simultaneously at the verbal and nonverbal levels. Verbal communication without nonverbal transmission is practically nonexistent. Thus, there are areas where the way information is communicated is of great importance not only for the quality of the future relationship such as patient and doctor but also to trigger a psychosomatic response, which can be translated into the functioning of the patient's immune system. Indeed, it should be pointed out that, for example, an appropriate tone of voice can allow for stress reduction when communicating a diagnosis. A study from 2011 funded by the National Cancer Institute shows that nonverbal information revealed in a lower tone of voice and a slower rate of speech gives the impression of being more empathetic [18]. This has a direct bearing on the patient's mental state, who feels better understood and embraced with compassion [56]. And although more research has been conducted within the realm of verbal empathic communication it is indicated that non-verbal based on the tone of voice and rate of speech is equally important [28]. How the message is conveyed is of particular importance important in the case of oncology patients, who face high levels of stress, tension, and fears for their lives as they face dealing with the disease [22].

Another study that confirms the importance of voice tone and tempo on levels of relaxation was conducted in 2006 in the biofeedback research laboratory of the Department of Behavioral Medicine and Psychiatry at West Virginia University. It investigated the effectiveness of progressive relaxation training (PRT) on selected vocal characteristics and its impact on the treatment process [15]. In the study [15], the goal was to see how the volume, pitch, timbre of the voice, and intensity of speech could affect the therapeutic process. As early as 1979, Ryan and Moses showed [26] that a soft, melodious voice can translate into treatment effectiveness. In addition to subjective assessment of the relaxation state or the subjects' perception of speech characteristics, participants in the experiment had their heart rate (HR) measured, and EMG signals were collected, verifying the electrical function of the electrical activity of muscles and peripheral nerves. The intensity of the voice conducting the relaxation training was measured in decibels, the tone of voice in Hz, and the number of syllables per second of tape was calculated. The results of the study clearly show that a voice that lowered and became more monotonous during the session caused a significant reduction in EMG levels (electromyography) which translated into a reduction in muscle tension. At the same time, the subjective feelings of the subjects confirmed that the way they used, and modulated their voice in therapy had an impact on their level of relaxation.

Muscle tension, like other vegetative autoregulatory processes (body temperature, heart rate, blood pressure, intestinal motility, etc.) sweat secretion cannot be controlled consciously. Since 1972, more than 1,500 articles have been published in professional publications on GSR (cutaneous-galvanic response) is considered the most popular method for studying the phenomena of human psychophysiological phenomena [3]. Although GSR is an ideal measure for tracking emotional arousal, it is unable to reveal the emotional valence i.e., the quality of the emotion. The true power of GSR unfolds when combined with other data sources to measure complex dependent variables and provide a complete picture of emotional behavior. Often, this test is performed in experiments involving games, or reactions to images or videos presented [32]. This opens the way to seek quantitative answers based on skin responses to further questions related to the mode of communication used in the VR treatment program so that the solution can best serve to reduce the patient's stress level and support relaxation, to enhance the healing processes.

In research on voice analysis during discussions of bad news in oncology [18], the author also states that no study analyzed verbal content, speech analysis, and other related nonverbal behavior, and notes that this is a desirable research topic. Most studies focus on listening to music and not the voice itself, these studies show that relaxation music (e.g. Bach, Vivaldi, Mozart) results in a slowing of the heart rate [6].

Music can strongly evoke and modulate emotions and mood, as well as changes in heart function, blood pressure (BP), and respiration. In the various studies on the effects of music on the heart, there is a wide variety of methods and quality, but can be established that: heart rate (HR) and respiratory rate (RR) are higher in response to exciting music compared to calming music [16]. W a study of music therapy to help treat children with cancer, music reduced pain ratings, heart rate, respiratory rate, and feelings of anxiety, during lumbar puncture, when children had headphones with music, they felt less pain and were calmer and relaxed during and after the procedure. All of these children wanted headphones with music when they next undergo the procedure [20]. This proves that listening to music already has its applications during medical therapies. There are not many publications that talk about listening to the voice itself, but studies show that people already in the womb can recognize the mother's voice, which has been observed to reduce the fetal heart rate [41]. Hypnosis (a recording with a slow breathing command) has also been shown to reduce heart rate, even in stressful situations such as dental procedures [6].

Thus, the VR treatment program's assumption that listening to a calming voice reduces heart rate appears to be true [23], and the creation of a device to monitor heart rate and attempt to lower it using a voiceover and guided relaxing trance will make it possible to study more closely how initially rhythmically spoken words at the same rate as the initial heartbeat rhythm heart rate affects the heart rate after the words are slowed down and whether the patient will calm down.

"The real power of understanding lies in not allowing our knowledge to be fettered by what we do not know." — stated Ralph Waldo Emerson — which is why it is important to continue research and see what combinations between the breath, the voice of the speaker, the rate of speech will bring the best effects thru VR treatment.

2 Experiment description

The EEG laboratory (Fig. 1) in the Department of Neuroinformatics and Biomedical Engineering at the Maria Curie-Skłodowska University in Lublin is equipped with apparatus that allows the precise study of bioelectrical changes occurring in the patient's brain thanks to an EGI dense matrix amplifier (Electrical

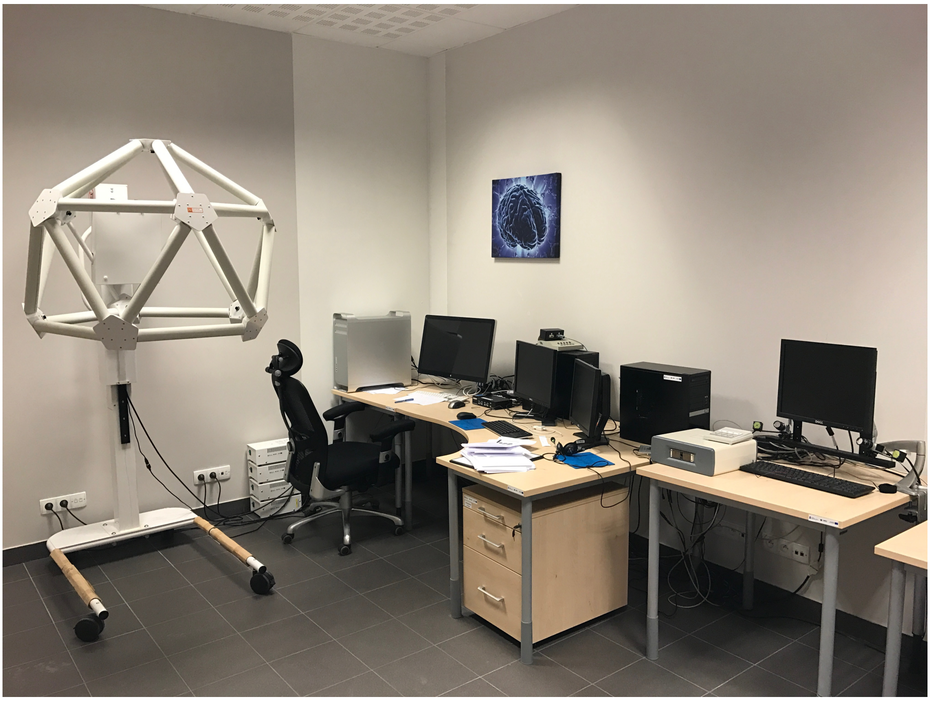


Figure 1: EEG laboratory in the Department of Neuroinformatics and Biomedical Engineering at Maria Curie-Sklodowska University in Lublin, Poland

Geodesic Systems, Oregon, USA), to which caps equipped with 256 electrodes (HydroCel GSN 130 Geodesic Sensor Nets) can be connected. The lab offers the ability to record signals at frequencies up to 1000 Hz (with simultaneous measurements by all 256 electrodes). The laboratory has a GPS photogrammetric station with GeoSource software that enables the application of source localization algorithms and the precise generation of a model of the subject's brain.

For a trial approach aimed at designing a research protocol to answer the formulated questions formulated by the Ordering Party, a relatively simple test was proposed involving putting the patient (in this case, the Department Head) into a state of deep relaxation by a qualified therapist (Katarzyna Zemla, M.Sc, SWPS doctoral student, Master of Cognitive Behavior BIK), and then recording the electrical activity of his brain in four main activity bands (alpha, beta, theta, delta) and measuring his heart rate and hemoglobin saturation throughout the entire study.

Hemoglobin saturation (SpO2) in the blood was measured using a Kermed A310 pulse oximeter. Heart rate was measured using a Xiaomi Mi Band 5. After the test, the patient was measured at a photogrammetric station. During the main part of the study, only the patient and the therapist were in the laboratory room, and twilight prevailed. The lab technicians were present only during the preparation of the patient and during the geodetic measurements. The study conducted on December 11, 2020, lasted 30 minutes.

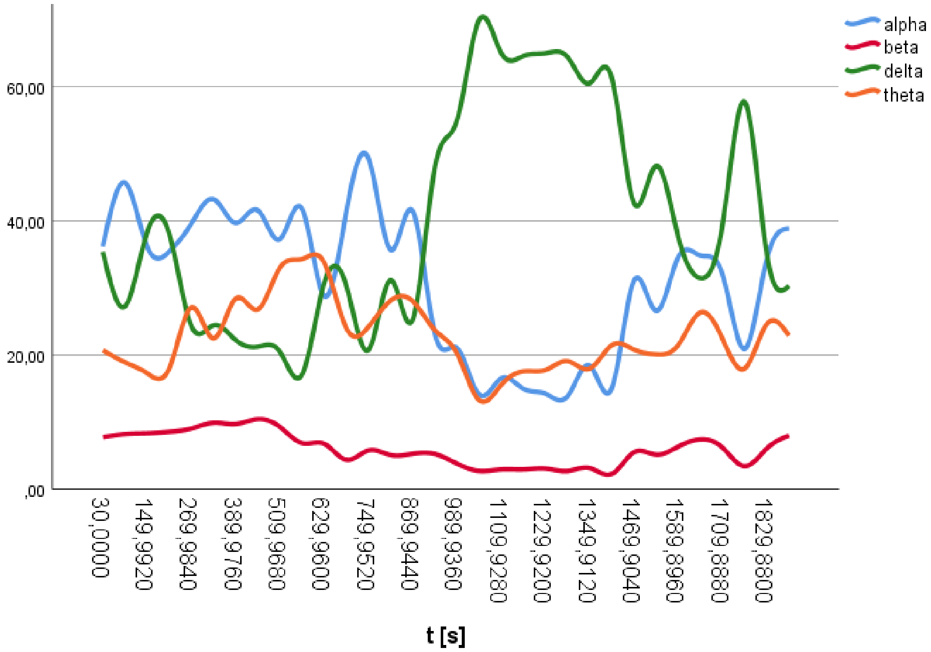


Figure 2: Percentage of each wave during the experiment

3 Experiment description

The percentage of each band of the electrical activity of the patient's cerebral cortex during the test is shown in Fig. 2. Changes in the subject's heart rate during the experiment are presented in Fig. 3. A graph of changes in hemoglobin saturation is shown in Fig. 4.

We can observe an increasing and then relatively high proportion of delta waves starting from about 870 seconds of the test (see. Fig. 2). This is accompanied by a relatively high proportion of alpha and theta waves at the beginning of the study with a low level of beta waves throughout the experiment. Starting at 870 seconds of the survey, there is a slight decrease in the contribution of alpha and theta waves as delta activity increases (see. Fig. 2).

The increase in delta activity is accompanied by an increase in pulse rate (see. Fig. 3) and a slight but observable increase in hemoglobin saturation (SpO2) (see. Fig. 4).

As is well known, the more than 50% contribution of delta waves to brain activity is associated with the phase of deep meditation or deep sleep. It can be presumed that an increase in the patient's delta brain activity above 60% in the study was related to the therapist's attempt to by the therapist to obtain the phenomenon of dreaming in sleep, which took place at that time. time. On the other hand, the increase in pulse rate may indicate a correlation between this phase with the visualizations occurring in the patient's state at the time of the

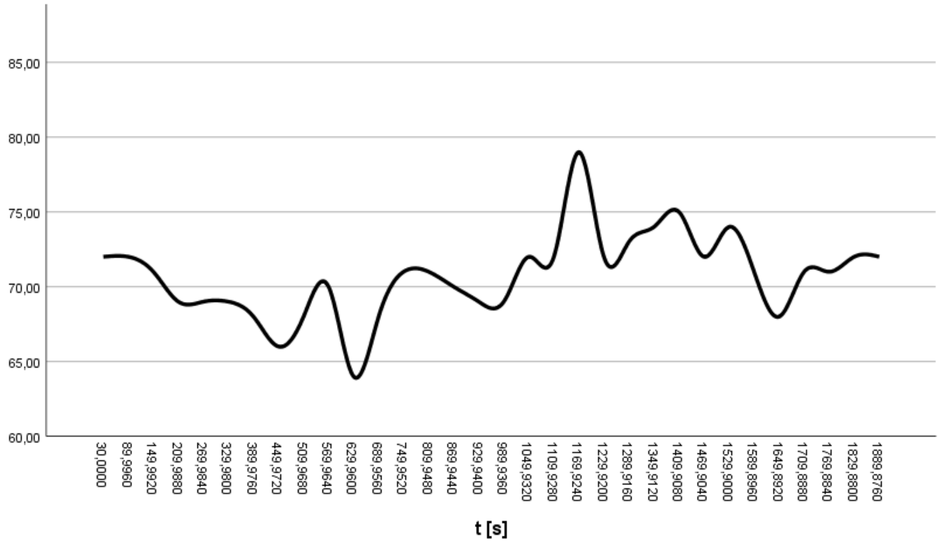


Figure 3: Pulse variations during the experiment

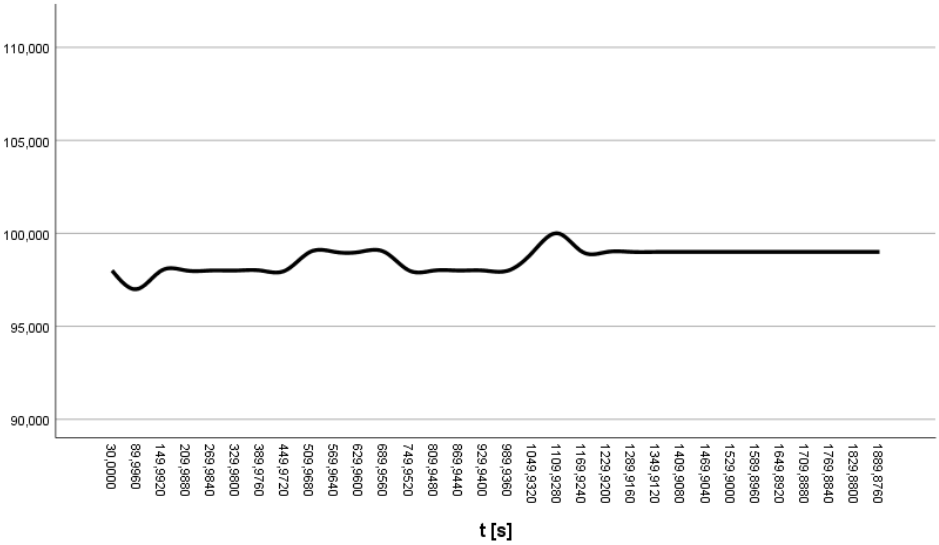


Figure 4: Hemoglobin saturation level (SpO2) during the experiment

examination. Theta waves at relatively high levels (about 30% on average) confirm the state of hypnosis into which the patient was put, also a state associated with shallow sleep; their decrease starting at 870 seconds indicates a rapid transition of the patient into a state resembling deep sleep.

It is difficult to conclude a single case. However the study pilot study was intended to show that we have the possibility of reliable quantitative recording of the electrical activity of the cerebral cortex and combined with a relatively simple to use and inexpensive apparatus, we can look for correlations of this activity with pulse rate and blood hemoglobin saturation (SpO2). After connecting the galvanometer it will be possible to study stress levels changes.

4 Recommendations

4.1 The rationale behind the VR treatment concept

The originators of the project rightly point out that the poor mental state in which most oncology patients find themselves reduces their quality of life during and after the various stages of treatment and can delay the processes of treatment and recovery. Therefore, the goal of the VR treatment program is to improve the mental state of patients so that they can experience positive emotional states even if they do not have the exceptional mental strength and are not able to control their thoughts and negative states. Patients by putting on the goggles and headphones could create new experiences and build positive beliefs and attitudes toward the healing and treatment process and reinforces and stabilizes a positive emotional state. In contrast, the poor mental state in which most oncology patients prolongs and impedes the treatment and recovery processes, and above all, reduce the quality of life during treatment. VR solution would allow the solution even if we face difficulty with access to specialized psycho-oncological help.

Carl Simonton's therapy, mentioned earlier, is based on activities in the following areas: behavior (relaxation, creating new habits), beliefs (changing unhealthy beliefs to ones that give us peace of mind and energy to act), emotions (maintaining hope, dealing with emotions that harm us, learning to cope with everyday stressful situations), spirituality, communication with supportive people (building a support system, learning healthy communication), and physicality (diet, movement, the role of play in the recovery process).

The many assumptions not only of selecting the most effective method but also of how to combine it with technology, which today offers the possibility of creating virtual worlds, cause many hypotheses and unknowns to arise, which need to be further investigated and verified. The relaxation module itself, for it to respond to changes in the patient's physiological state as well as the therapist must be designed to respond to his breathing, pulse, or measuring changes in the skin's electrical resistance.

When relaxation is led by a therapist, he or she often sees and adjusts the guidance of the body relaxation and visualization to the patient's breathing. The hypnotherapist can see when the patient's chest is on the inhale and lowers on the exhale. So the open question remains how to map this alignment when the patient puts on the goggles and receives instructions from the VR treatment program in the most effective way?

4.2 Recommended research

To extend current phase of conducted pilot study it is recommended to proceed with further steps such as:

- In-depth research on the susceptibility of patients to relaxation intervention depending on a set of variables obtained from psychological questionnaires: anxiety, depressiveness, introverted, extroverted personality types which may determine natural attitude toward diagnosis.
- Conducting experiments to build classifiers capable of suggesting the most appropriate pace and method of conducting the relaxation intervention.
- Conducting experiments to test the performance of the constructed clarifiers.

This is the initial stage of our project.

Depending on the personal properties and external influence each patient can have an individual ability to be exposed to relaxation, varying in time and other conditions.

In future, it will be useful to investigate the pace at which particular subjects get into a deep state of relaxation. It was only our expectation that they ought to do this in around 14 minutes. However, each individual can be characterised and most probably is by his own pace. Plotting their state in the function of time would be recommended.

Using machine learning classifiers is expected to find application in the classification of biomedical signals towards therapy support citedylkag2021pilot, mikolajewska2014non. Machine learning tools and algorithms have been used for decades also for lots of disorders diagnostics like alcoholism or depression [27, 12] and others [47] using new measures like those defined in [46] as well as advanced modelling of biological systems behaviour [35] including diagnostics purposes [13, 44, 45].

Acknowledgements

Research presented herein was supported by the City of Lublin as a part of cooperation of Lublin City Hall and Maria Curie-Sklodowska Univeristy towards pilot study for VRMed project. The Authors would like to thank Ms. Ewa Lewandowska and Mr. Marek Rudziński of VRMed for fruitful discussions, inspiration and possibility of working together on new ideas.

References

[1] JE Auerbach, TD Oleson, and GF Solomon. Biofeedback, guided imagery, and hypnosis as an adjunctive treatment for aids and aids-related complex. In *Proceedings o f the Third National Conference on the Psychology of Health,*

Immunity, and Disease in Storrs, CT: The National Institute for the Clinical Application of Behavioral Medicine, volume 50, 1992.

[2] Gary G Berntson, John T Cacioppo, and Karen S Quigley. Respiratory sinus arrhythmia: autonomic origins, physiological mechanisms, and psychophysiological implications. *Psychophysiology*, 30(2):183–196, 1993.

[3] Wolfram Boucsein. *Electrodermal activity.* Springer Science & Business Media, 2012.

[4] P Bušek and D Kemlink. The influence of the respiratory cycle on the eeg. *Physiol Res*, 54:327–33, 2005.

[5] Bruce M Cappo and David S Holmes. The utility of prolonged respiratory exhalation for reducing physiological and psychological arousal in non-threatening and threatening situations. *Journal of psychosomatic research*, 28(4):265–273, 1984.

[6] Gianfranco Cervellin and Giuseppe Lippi. From music-beat to heart-beat: a journey in the complex interactions between music, brain and heart. *European journal of internal medicine*, 22(4):371–374, 2011.

[7] K Davies. Search and deploy. bio–it world, october 16, 2006, 2006.

[8] Fawzy I Fawzy, Nancy W Fawzy, Christine S Hyun, Robert Elashoff, Donald Guthrie, John L Fahey, and Donald L Morton. Malignant melanoma: Effects of an early structured psychiatric intervention, coping, and affective state on recurrence and survival 6 years later. *Archives of general psychiatry*, 50(9):681–689, 1993.

[9] Howard S Friedman. *Encyclopedia of mental health.* Academic Press, 2015.

[10] Sarnik Gaurav, Sinha Meenakshi, Ghate Jayshri, and Sinha Ramanjan. Effect of alterations in breathing patterns on eeg activity in normal human subjects. *Int J Curr Res Med Sci*, 2:38–45, 2016.

[11] John H Gruzelier, Trevor Thompson, Emma Redding, Rosemary Brandt, and Tony Steffert. Application of alpha/theta neurofeedback and heart rate variability training to young contemporary dancers: State anxiety and creativity. *International Journal of Psychophysiology*, 93(1):105–111, 2014.

[12] Erwin Roy John, LS Prichep, J Fridman, and P Easton. Neurometrics: computer-assisted differential diagnosis of brain dysfunctions. *Science*, 239(4836):162–169, 1988.

[13] Aleksandra Kawala-Sterniuk, Michal Podpora, Mariusz Pelc, Monika Blaszczyszyn, Edward Jacek Gorzelanczyk, Radek Martinek, and Stepan Ozana. Comparison of smoothing filters in analysis of eeg data for the medical diagnostics purposes. *Sensors*, 20(3):807, 2020.

[14] PJ Kelly. Evaluation of a meditation and hypnosis-based stress management program for men with hiv. In *Poster presented at the Fifth International Conference on AIDS, Montreal*, 1989.

[15] Glenn E Knowlton and Kevin T Larkin. The influence of voice volume, pitch, and speech rate on progressive relaxation training: application of methods from speech pathology and audiology. *Applied psychophysiology and biofeedback*, 31(2):173–185, 2006.

[16] Stefan Koelsch and Lutz Jäncke. Music and the heart. *European heart journal*, 36(44):3043–3049, 2015.

[17] Margaret McCoy Lynch. Factors influencing successful psychotherapy outcomes. *Master of Social Work Clinical Research Papers*, 2012.

[18] Monica McHenry, Patricia A Parker, Walter F Baile, and Renato Lenzi. Voice analysis during bad news discussion in oncology: reduced pitch, decreased speaking rate, and nonverbal communication of empathy. *Supportive Care in Cancer*, 20(5):1073–1078, 2012.

[19] P Newton and L Marx. *The Clinical Use of Hypnotic Imagery in HIV Positive Men.* The Center, 1990.

[20] Thanh Nhan Nguyen, Stefan Nilsson, Anna-Lena Hellström, and Ann Bengtson. Music therapy to reduce pain and anxiety in children with cancer undergoing lumbar puncture: a randomized clinical trial. *Journal of Pediatric Oncology Nursing*, 27(3):146–155, 2010.

[21] Burkhard Peter. Hipnoza i psychoterapia pacjentów z hiv, arc i aids. hypnosis and psychotherapy with hiv, arc and aids patients. *Psychiatria polska*, 1(23), 2005.

[22] JT Ptacek and Tara L Eberhardt. Breaking bad news: a review of the literature. *Jama*, 276(6):496–502, 1996.

[23] A Ramírez-Carrasco, C Butrón-Téllez Girón, O Sanchez-Armass, and M Pierdant-Pérez. Effectiveness of hypnosis in combination with conventional techniques of behavior management in anxiety/pain reduction during dental anesthetic infiltration. *Pain Research and Management*, 2017, 2017.

[24] Ernest Lawrence Rossi. *The Breakout Heuristic: The New Neuroscience of Mirror Neurons, Consciousness, and Creativity in Human Relationships.* Milton H. Erickson Foundation Press, 2007.

[25] Marc A Russo, Danielle M Santarelli, and Dean O'Rourke. The physiological effects of slow breathing in the healthy human. *Breathe*, 13(4):298–309, 2017.

[26] Victor L Ryan and James A Moses. Therapist warmth and status in the systematic desensitization of test anxiety. *Psychotherapy: Theory, Research & Practice*, 16(2):178, 1979.

[27] Nilima Salankar, Saeed Mian Qaisar, Paweł Pławiak, Ryszard Tadeusiewicz, and Mohamed Hammad. Eeg based alcoholism detection by oscillatory modes decomposition second order difference plots and machine learning. *Biocybernetics and Biomedical Engineering*, 2022.

[28] Klaus R Scherer, Tom Johnstone, and Gundrun Klasmeyer. *Vocal expression of emotion.* Oxford University Press, 2003.

[29] O Carl Simonton and STEPHANIE MATTHEWS-SIMONTON. Cancer and stress counselling the cancer patient. *Medical Journal of Australia*, 1(13):679–683, 1981.

[30] O Carl Simonton, Stephanie Matthews-Simonton, and T Flint Sparks. Psychological intervention in the treatment of cancer. *Psychosomatics*, 21(3):226–233, 1980.

[31] Jonathan C Smith. *Relaxation, meditation, & mindfulness: A mental health practitioner's guide to new and traditional approaches.* Springer Publishing Company, 2005.

[32] Mohammad Soleymani, Sadjad Asghari-Esfeden, Maja Pantic, and Yun Fu. Continuous emotion detection using eeg signals and facial expressions. In *2014 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2014.

[33] David Spiegel, HelenaC Kraemer, JoanR Bloom, and Ellen Gottheil. Effect of psychosocial treatment on survival of patients with metastatic breast cancer. *The lancet*, 334(8668):888–891, 1989.

[34] E Szeląg. Nowe tendencje w terapii logopedycznej w świetle badań nad mózgiem,[w:] podstawy neurologopedii. *Podręcznik akademicki, red. T. Gałkowski, G. Jastrzębowska, E. Szeląg, Opole*, pages 1028–1048, 2005.

[35] Ryszard Tadeusiewicz. Neural networks as a tool for modeling of biological systems. *Bio-Algorithms and Med-Systems*, 11(3):135–144, 2015.

[36] Michal Teplan, Anna Krakovská, and Marián Špajdel. Spectral eeg features of a short psycho-physiological relaxation. *Measurement Science Review*, 14(4):237–242, 2014.

[37] Frederick Travis. Autonomic and eeg patterns distinguish transcending from other experiences during transcendental meditation practice. *International Journal of psychophysiology*, 42(1):1–9, 2001.

[38] Julian Treasure. *Sound business.* Management Books 2000 Limited, 2011.

[39] Ilse Van Diest, Karen Verstappen, André E Aubert, Devy Widjaja, Debora Vansteenwegen, and Elke Vlemincx. Inhalation/exhalation ratio modulates the effect of slow breathing on heart rate variability and relaxation. *Applied psychophysiology and biofeedback*, 39(3-4):171–180, 2014.

[40] Elke Vlemincx, Ilse Van Diest, Steven De Peuter, and Omer Van den Bergh. Relaxatie: een 'cold case'hervat. *Psychologie en Gezondheid, jaargang 2007*, 35(1):14–23, 2007.

[41] Kristin M Voegtline, Kathleen A Costigan, Heather A Pater, and Janet A DiPietro. Near-term fetal response to maternal spoken voice. *Infant Behavior and Development*, 36(4):526–533, 2013.

[42] PM West, EM Blumberg, and FW Ellis. An observed correlation between psychological factors and growth rate of cancer in man. In *Cancer Research*, volume 12, pages 306–307. AMER ASSOC CANCER RESEARCH PO BOX 11806, BIRMINGHAM, AL 35202, 1952.

[43] Mariusz Wirga and Michael DeBernardi. The abcs of cognition, emotion, and action. *Archives of Psychiatry and Psychotherapy*, 4(1):5–16, 2002.

[44] Grzegorz M Wójcik, Andrzej Kawiak, Lukasz Kwasniewicz, Piotr Schneider, and Jolanta Masiak. Azure machine learning tools efficiency in the electroencephalographic signal p300 standard and target responses classification. *Bio-Algorithms and Med-Systems*, 15(3), 2019.

[45] Grzegorz Marcin Wojcik, Joalnta Masiak, Andrzej Kawiak, Lukasz Kwasniewicz, Piotr Schneider, Nikodem Polak, and Anna Gajos-Balinska. Mapping the human brain in frequency band analysis of brain cortex electroencephalographic activity for selected psychiatric disorders. *Frontiers in Neuroinformatics*, 12:73, 2018.

[46] Grzegorz Marcin Wójcik, Joalnta Masiak, Andrzej Kawiak, Piotr Schneider, Lukasz Kwasniewicz, Nikodem Polak, and Anna Gajos-Balinska. New protocol for quantitative analysis of brain cortex electroencephalographic activity in patients with psychiatric disorders. *Frontiers in Neuroinformatics*, 12:27, 2018.

[47] Grzegorz Marcin Wójcik, Jolanta Masiak, Andrzej Tadeusz Kawiak, Lukasz Kamil Kwasniewicz, Piotr Schneider, Filip Postepski, and Anna Gajos-Balinska. Analysis of decision-making process using methods of quantitative electroencephalography and machine learning tools. *Frontiers in Neuroinformatics*, 13:73, 2019.

Network of Social Contacts in Poland of the Times of SARS-CoV-2 Pandemic

Andrzej Krajka[*]
Mateusz Mackiewicz

1 Introduction

The third year of the SARS-CoV-2 pandemic has passed and most research effort is devoted to studying the impact of the pandemic on social relations, the economy, politics and other aspects of the life of people in Poland. It is known that social contacts were dramatically restricted (or moved to the Internet), independently of lockdown decisions. In this paper, we would like to note the changes in social behaviour in Poland in terms of network parameters such as assortativity (homophily), mean distance and the law of degree distribution. These values are computed from simulations of the course of the epidemic on different networks and then compared with the observed runs of the epidemic.

2 Data sources

Simulations of the course of the epidemic will be carried out on the following seven graphs:

DOLPHIN The file *dolphins.gml* [13] contains an undirected social network of frequent associations between 62 dolphins in a community living off Doubtful Sound, New Zealand, as compiled by Lusseau et al. (2003) [5]. See additionally [4] and [3].

KARATE The file *karate.gml* contains a network of friendships between 34 members of a karate club at a US university, as described by Wayne Zachary in 1977 [12]. The data were taken from [15].

LUX The graph of roads in Luxembourg taken from [10].

[*] Corresponding author — andrzej.krajka@poczta.umcs.pl
This article is a part of Mateusz Mackiewicz's Master Thesis prepared in the Insitute of Computer Science, UMCS.

FACEBOOK This dataset consists of *circles* (or *friend lists*) from Facebook. Facebook data were collected from survey participants using the Facebook app. The dataset includes node features (profiles), circles, and ego networks. Facebook data have been anonymized by replacing the Facebook-internal ids for each user with a new value. Also, while feature vectors from this dataset have been provided, the interpretation of those features has been obscured. For instance, where the original dataset may have contained a feature "political=Democratic Party", the new data would simply contain "political=anonymized feature 1". Thus, by using the anonymized data it is possible to determine whether two users have the same political affiliations, but not what their political affiliations represent. The data were taken from [14].

FOAF The FOAF graph was taken from [16]. FOAF (an acronym for "friend of a friend") is a machine-readable ontology describing persons, their activities and their relations to other people and objects. The FOAF language (based on RDF and XML) specification may be found in [17]. In our investigation, we restrict ourselves to triples with the properties *foaf:knows*, only. FOAF allows groups of people to describe social networks without the need for a centralised database.

BINOM The BINOM graph is random and was produced by the program:

```
n<-50
p=0.1
k=4

wyn<-c(rbinom(n*n,k,p))
wyn<-matrix(as.numeric(wyn>0),n,n)

rownames(wyn)<-colnames(wyn)<-seq(1,n)
library(igraph)
Graf2<-graph_from_adjacency_matrix(wyn)
```

GO The Gene Ontology (GO) knowledge base is the world's largest source of information on the functions of genes. This knowledge is both human-readable and machine-readable and is a foundation for computational analysis of large-scale molecular biology and genetics experiments in biomedical research. Our network GO was produced by the program:

```
library(topGO)
library(org.Hs.eg.db)        # Human

# Human gene list library(org.Hs.eg.db)
genlist<-names(as.list(org.Hs.egALIAS2EG))
golist<-as.data.frame(GOTERM)
il=length(genlist)

set.seed(111)
genes=rep(0,il)
names(genes)=sample(genlist, il)
```

```
selection <- function(allScore){
    # function that returns TRUE/FALSE for p-values<0.05
    return(allScore < 0.5)}
allGO2genes <- annFUN.org(whichOnto="BP",
                          feasibleGenes=NULL,
                          mapping="org.Hs.eg.db",
                          ID="symbol")
GOdata <- new("topGOdata",
              ontology="BP",
              allGenes=genes,
              annot=annFUN.GO2genes,
              GO2genes=allGO2genes,
              geneSel=selection,
              nodeSize=3)

library(igraph)
Graf1 <- graph_from_graphnel(GOdata@graph)
```

3 Data properties

Main properties of the above defined graphs are summarized in Table 1. For the social network analysis, we will consider two values:

Law distribution We consider two laws of distribution:

Power law of distribution $P_k \sim k^{-\alpha}, k = 1, 2, \ldots$

and

Exponential distribution $P_k \sim e^{-k/\alpha}, k = 1, 2, \ldots$

where P_k is the number of vertex of a degree k in the graph. The type of a given graph is determined based on a multiple R-Squared, which determines how well your model fits the data. A comparison of multiple R-squared together with the α coefficients of the considered graphs is given in Table 2.

Assortative mixing The values introduced and described in [9] and [8] and computed for our graphs in Table 2. It is a bias in favour of connections between network nodes with similar characteristics. In the specific case of social networks, assortative mixing is also known as homophily.

Table 1: Basic properties of the considered graphs

Graph $\mathcal{G}$	Mean degree	Account of vertex	Account of edges	Is simple?	Is consistent?	Diameter	Centrali- zation
DOLPHIN	5.13	62	159	No	Yes	8	0.1126
KARATE	4.59	34	78	No	Yes	5	0.3761
LUX	2.09	114599	119666	Yes	Yes	1337	$1.71 \cdot 10^5$
FACEBOOK	43.69	4039	88234	Yes	Yes	8	0.1240
FOAF	1.95	34061	33223	Yes	No	17	0.0468
BINOM	40.28	50	1007	Yes	Yes	2	0.1429
GO	4.52	11455	25883	Yes	Yes	16	0.0043

Table 2: Additional properties of the considered graphs.

Graph	Power		Exponential		Vertex with	Density	Mean	Assorta-
$\mathcal{G}$	R^2	α	R^2	α	$deg > 10$		distance	tivity
DOLPHIN	0.008	0.074	0.017	0.005	4.8%	0.084	$3.4 \notin (1.4, 2.9)$	-0.04360
KARATE	0.121	0.268	0.002	0.003	8.8%	0.139	$2.4 \in (1.3, 2.8)$	-0.47561
LUX	0.016	0.024	0.026	9e-7	0.0%	9.1e-6	$54.9 \notin (2.5, 4.7)$	0.34186
FACEBOOK	0.025	-0.189	0.001	-2e-5	76.2%	0.005	$4.3 \notin (2.1, 3.9)$	-0.04205
FOAF	0.003	0.022	0.009	4e-6	1.1%	2.9e-5	$1.2 \notin (2.3, 4.4)$	-0.15067
BINOM	0.003	3e-5	0.001	1e-6	100.0%	0.411	$1.7 \notin (1.9, 3.6)$	-0.00409
GO	0.002	0.012	0.003	6e-6	4.5%	1.9e-4	$3.6 \in (2.2, 4.2)$	-0.06215

In literature, the conditions for graphs describing social networks (in normal times) are:

(a) Rather power (c.f. [11]) than exponential distribution (KARATE, FACEBOOK, BINOM).

(b) Density between 0.1 and 0.25 (BINOM, KARATE)

(c) Big homophily (the great values of assortativity) (LUX, BINOM, FACEBOOK, DOLPHIN)

(d) Users with degree greater then 10 should be greater then 35% (FACEBOOK, BINOM)

(e) Mean distance should be small (less than 5) (FOAF, BINOM, KARATE, DOLPHIN, GO, FACEBOOK). In the [2] it is evaluated that the mean distance for social networks with N vertex should belong to $(\ln \ln N, \frac{\ln N}{\ln \ln N})$ interval (KARATE, GO).

Thus, the best network describing the social contacts between people should be BINOM (it isn't a social network, it is a random graph) and a bit worst KARATE (it is a real social network) and FACEBOOK (it is a real social network too). There is interesting the far position of another social network FOAF. The number of satisfying points for each graph we show in six columns of Table 3.

4 Methods

This ranking (described at the end of the previous section and deals with "normal" times) should be verified by simulation of the runs of SARS CoV 2 epidemic in Poland, considering the above graphs as the graph of social contacts of Polish people. We exactly make:

- We take the COVID data for Poland from [18] and proceed according to the described in [19] to obtain optimal β and γ values for the SIR model. After that, we smooth, using the Kalman filter, the R function according to the method described in [1]. Thus, we obtain a "real" effective reproductive function $\widehat{R}(t)$.

- For each graph $\mathcal{G}$, under different levels of mortality (μ) and contagiousness (λ), we simulate the course of the epidemic, obtaining the "simulated" effective reproductive function $R_{\lambda,\mu,\mathcal{G}}(t)$.

- Using the time series distances *dist*: Dynamic Time Wrapping DTW and Short Time Series Distance STS we find the value

$$Err(dist, \mathcal{G}) = \inf_{\beta,\lambda>0} dist(R_{\lambda,\mu,\mathcal{G}}, \widehat{R}), \quad dist \in \{DTW, STS\}.$$

  These values are summarized in Table 3. It is worth stressing that we use a shape-based distance instead that based on feature, structure or prediction, because of the short time of simulation (100 days) and the "small" number of vertexes ("people") for some considered graphs. The time series distances are implemented in the *TSDist* library in R, and are described in detail in [7] and [6].

4.1 Input data

Let us recall the definition of truncated exponential law $E(\alpha, N)$:

$$P[E(\alpha, N) = k] = \begin{cases} \frac{\alpha^{k-1}}{(k-1)!}e^{-\alpha}, & \text{for } k = 1, 2, \dots N-1, \\ 1 - \sum_{i=0}^{N-2} \frac{\alpha^i}{i!}e^{-\alpha}, \text{ for } k = N, & \\ 0, & \text{otherwise .} \end{cases}$$

Input data:

L The time of duration of the disease

$\{p_k, k = 1, 2, \dots L\}$ The probability of infection of other person in the kth day of illness. We will assume that the probability of infection has law $E(\lambda, L)$ for some $\lambda > 0$.

$\{death_k, k = 1, 2, \dots L\}$ The probability of death of the infected person in the kth day of illness. Similarly, as above, this distribution is given by $E(\mu, L)$ for some $\mu > 0$.

t_{car} The carention time after illness when person cannot get infected.

T_{max} The simulation time (in days).

$graph$ The graph of interpersonal contacts.

x_o The vector of persons who are initially infected.

4.2 Output

wyn_1 The result matrix $wyn_1[t, k], 1 \leq t \leq T_{max}, 0 \leq k \leq L$, is the number of ill persons in tth moment who in kth day of illness ($k = 1, 2, \dots L$) whereas $wyn_1[t, 0]$ is the number of health or recover persons.

wyn_2 The structure of persons $wyn_2[t, k], , 1 \leq t \leq T_{max}, 0 \leq k \leq 4$ who are not yet infected ($k = 0$) or who were infected kth times ($k = 1, 2,$) or who were infected three times at least for $k = 3$ or dead for $k = 4$.

wyn_3 The SIR model $wyn_3[t,k], 1 \leq t \leq T_{max}, 1 \leq k \leq 3$ — Suspected, Infected and Recovered for k=1,2,3, respectively.

$R_{\lambda,\mu,\mathcal{G}}$ The value of $R_{\lambda,\mu,\mathcal{G}}$ obtained from the simulation:

```
Isim <- cumsum(wyn3[, 2])
Rsim <- ts(Isim[2:dane$Tmax]/Isim[1:(dane$Tmax-1)])
```

4.3 Algorithm

The main loop is over the graphs $\mathcal{G}$ and for λ and μ. For every time $t \in [1, T_{max}]$ we make the following:

- For every person (vertex) who is ill in $k \in [1, L]$ time of illness, we draw with a probability $death_k$ if it this person dies, and the mark this person as dead.
- For each ill person we increase by 1 their time of illness (if the time is greater than L, we set time to $-tcar$).
- For each ill person with a positive time of illness we randomly choose (with probability p_k) their neighbour (in sense graph $\mathcal{G}$) who is neither ill nor dead, and then we mark this neighbour as ill in the first day of illness.
- We update the output tables $wyn_i, i = 1, 2, 3$.

and after the simulation we compute $R_{\lambda,\mu,\mathcal{G}}$ and $dist(R_{\lambda,\mu,\mathcal{G}}, \widehat{R})$, $dist \in \{DTW, STS\}$. In the loop of λ and μ, we minimize both this distances computing $Err(dist, \mathcal{G})$ finding optimal values λ and μ.

5 Results

The sixth column is obtained by ranking the number of points satisfying the "normal times" social networks, whereas the seventh one is the average of DTW and STS distances rankings (from worst to best). "In normal times" the best are graphs

Table 3: Simulation results

Graph $\mathcal{G}$	λ	μ	Err DTW	Err STS	Rank in normal times	Rank in COVID times
DOLPHIN	0.8	1.1	68.0260	1.6082	1	4
KARATE	0.2	0.9	68.5272	1.6447	3	3
LUX	0.8	1.0	87.1263	1.2726	1	4
FACEBOOK	0.2	0.7	46.7006	1.8674	3	4
FOAF	0.4	0.3	80.3302	1.3590	0	4
BINOM	0.6	0.3	66.4789	1.7893	4	3.5
GO	0.7	1.2	52.8516	1.4853	1	5.5

BINOM and a bit worst KARATE and FACEBOOK with a small diameter, mean centralization, big mean degree and mean assortativity, whereas "in the pandemic times" the best is the graph GO with a big diameter, very small centralization, small mean degree and a little smaller assortativity. Thus the contacts in "pandemic times" are bounded (smaller degree) and scattered (smaller centralization) but are wider (bigger diameter).

References

[1] Arroyo-Marioli, F., Bullano, F., Kucinskas, S. & Rondón-Moreno, C. (2021). Tracking R of COVID-19: A new real-time estimation using the Kalman filter. PloS one, 16(1), e0244474. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0244474

[2] Cohen, R. & Havlin, S. (2003). Scale-free networks are ultrasmall. Physical Review Letters, 90(5), 058701.

[3] Lusseau D., Evidence for social role in a dolphin social network, Preprint q-bio/0607048 (http://arxiv.org/abs/q-bio.PE/0607048)

[4] Lusseau D., The emergent properties of a dolphin social network, Proc. R. Soc. London B (suppl.) 270, S186-S188 (2003).

[5] Lusseau D., Schneider K., Boisseau O. J., Haase P., Slooten E., and Dawson S. M., The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations, Behavioral Ecology and Sociobiology 54, 396–405 (2003).

[6] Montero, P. & Vilar, J. A. TSclust: An R package for time series clustering. Journal of Statistical Software, (2015). **62,** 1–43.

[7] Mori U., Mendiburu A., Lozano J. A., Distance Measures for Time Series in R: The TSdist Package. R Journal, (2016), **8.2:** 451–459.

[8] Newman, M. E. J., Assortative Mixing in Networks, Physical Review Letters, 89, 20 (2002), American Physical Society (APS), http://dx.doi.org/10.1103/PhysRevLett.89.208701.

[9] Newman, M. E. J., Mixing patterns in networks, Physical Review E, 67, 2, (2003), American Physical Society (APS), http://dx.doi.org/10.1103/PhysRevE.67.026126.

[10] Ryan A., Rossi and Nesreen K. Ahmed, The Network Data Repository with Interactive Graph Analytics and Visualization, AAAI, https://networkrepository.com, (2015).

[11] Takac, L. & Zabovsky, M. (2012, May). Data analysis in public social networks. In International scientific conference and international workshop present day trends of innovations (Vol. 1, No. 6). Poland: Present Day Trends of Innovations Lamza.

[12] W. W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research 33, 452–473 (1977).

[13] https://github.com/csdashes/GraphStreamCommunityDetection/blob/master/data/dolphins.gml

[14] https://snap.stanford.edu/data/ego-Facebook.html

[15] https://github.com/oseledets/nla2015/blob/master/lectures/karate.gml

[16] https://ebiquity.umbc.edu/resource/html/id/82/foafPub-dataset

[17] http://xmlns.com/foaf/spec/

[18] https://data.europa.eu/euodp/en/data/dataset/covid-19-coronavirus-data

[19] https://statsandr.com/blog/covid-19-in-belgium/

SARS-CoV-2 Epidemic in Poland and Other Countries

Andrzej Krajka*
Sandra Kamińska

1 Introduction

The SARS-CoV-2 epidemic has been going on long enough to ask the question of which strategy to fight COVID proved the best: the "lockdown", the (preferred in Asia) "zero tolerance" strategy, balanced testing and quarantine strategy or another. The specialistic discussion overlaps the "political interest" and in consequence spoofs the statistics. In this paper, we don't judge which strategy was the best but on the base available data, we compare the course of the epidemic in Poland and other countries. The base of comparison will be the commonly used Kermack-McKendrick SIR Model. In restriction degree (due to limited input data) we indicate factors which mainly influenced the intensity of the epidemic.

2 Model SIR

The SIR model [12] is one of the most basic models for describing the temporal dynamics of an infectious disease in a population. It compartmentalizes people into one of three categories: those who are Suspected to be ill, those who are currently Infected, and those who have Recovered (with immunity). The SIR model can be described using a set of equations that describe the number (or proportion) of people in each compartment at every point in time. For the number of people we have

$$\begin{aligned}\frac{dS(t)}{dt} &= -\beta I(t)S(t),\\ \frac{dI(t)}{dt} &= \beta I(t)S(t)-\gamma I(t),\\ \frac{dR(t)}{dt} &= \gamma I(t), t\geq 0.\end{aligned} \tag{1}$$

*Corresponding author — andrzej.krajka@poczta.umcs.pl
This article is a part of Sandra Kamińska's Master Thesis prepared in the Insitute of Computer Science, UMCS.

whereas for proportions we should replace $S(t), I(t), R(t)$ by $S(t)/N, I(t)/N, R(t)/N$, respectively, where N denotes the population of a considered country. Here β is the average number of contacts per person per time, multiplied by the probability of disease transmission in contact between a susceptible and an infectious subject whereas γ is the probability of an infectious individual recovering in any time interval dt is simply γdt. If an individual is infectious for an average period D, then $\gamma = \frac{1}{D}$. This is also equivalent to the assumption that the length of time spent by an individual in the infectious state is a random variable with an exponential distribution.

The system 1 is non-linear, however it is possible to derive its analytic solution in implicit form [11]. It is possible to show that

$$S(t) = S(0)e^{-R_o(R(t)-R(0))}, t \geq 0, \tag{2}$$

where

$$R_o = \frac{\beta}{\gamma},$$

is the so-called basic reproduction number (also called basic reproduction ratio). This ratio is derived as the expected number of new infections (these new infections are sometimes called secondary infections) from a single infection in a population where all subjects are susceptible. It is the key value for the behaviour of an epidemic; when $R_o > 1$ the infection the number of infected persons increases, but not if $R_o < 1$.

3 Data sources

The source data of COVID history, excess mortality, stringency index [6], size of population and population density was pulled out from [1] (in *csv* format), although there are a lot of numerous websites with COVID information [2, 3, 4]. We read and later prepare data in an R language data frames format into two data frames:

dan Containing country name, date of observations, number of confirmed/death/recovered/active in this day, cumulative sums confirmed_cum/death_cum/recovered_cum/active_cum, which leads to

$$\begin{aligned} \widehat{I}(t) &= active\_cum(t), \\ \widehat{R}(t) &= death\_cum(t) + recovered\_cum(t), \\ \widehat{S}(t) &= N - \widehat{I}(t) - \widehat{R}(t) \end{aligned}$$

where N is the population size of the country. These fields of data frame are supplemented by the computed in time of analysis fields Re, Re_LCL, Re_UCL and $Reo(\widehat{R}_e)$ — the values of smoothing (Re) and unsmoothing (Reo) effective reproductive function. It is worth mentioning that the all analysis of the epidemic started at moment $u = \min\{t : \widehat{I}(t) \geq 1, \widehat{R}(t) \geq 1\}$ and was ended in the last available observation T (note that the interval $rang = [u, T]$ may be different for different countries).

df This is the characteristic of every considered country and contains such information as country, $T - t$, N — population size, excess deaths, stringency index, population density, the number of doctors per 100000 residents, climate_moderate, climate_subtropical, climate_tropical (the boolean values indicate the climatic zone of a considered country), and further computed values: $\beta, \gamma, v0, w0, meanReo, sdReo, meanRe, sdRe, sdRe\_Reo, shapRe\_Reo$.

4 Programs

After described above prepare input data, we done computations in some steps:

Step 1. Using *optim* from the library *deSolve*, we fit the SIR model (1) to our data $(\widehat{I}(t), t \in [u, T])$ by finding the values for β and γ that minimize the residual sum of squares $\sum_{t=u}^{T}(I(t) - \widehat{I}(t))^2$. For some countries (such as the Vatican City or Vanatu), the interval $[u, T]$ was too small (or even empty) thus, as a result, we produce NA values. Additionally, for Poland, we consider the solution for β and γ obtained from increasing data $\widehat{I}(s), s \in [u, t], t = u+5, u+6, \dots T$. Thus, we get some viewpoint on epidemic changes in Poland over times (cf. Figure 1).

Step 2. We compute

$$\begin{aligned} grI(t) &= \frac{\widehat{I}(t+1) - \widehat{I}(t)}{\widehat{I}(t)}, \\ Reo(t) &= \widehat{R}_e(t) = grI(t) + 1, \quad t \in [u, T-1]. \end{aligned} \tag{3}$$

Now, we use the so called Kalman filter to smooth $\widehat{R}_e$:

```
library(dlm)
model <- dlm(FF = gamma, V=v0, GG=1, W=w0, m0 = 4, C0=1e2)
      wynx <- dlmSmooth(grI, model)
      mse.list=dlmSvd2var(wynx$U.S, wynx$D.S)

      S=c(S1,S)
      se = 1.96*t(sapply(mse.list,
          FUN = function(x) sqrt(diag(x)))*sqrt(N/S))

      df$Re[rang]<-(wynx$s+1)*N/S
      df$Re_UCL[rang]=(wynx$s+1)*N/S+se[1,]
      df$Re_LCL[rang]=(wynx$s+1)*N/S-se[1,]
      df$Reo[rang]=(c(0, grI)+1)*N/S
```

for a theoretical description cf. [10]. The values $v0$ and $w0$ were computed by the genetic algorithm methods from the above procedure such that could be minimized the standard deviation error between $R_e(t) = R_{e,FF,v,GG,w,m0,C0}(t)$ and $\widehat{R}_e(t)$ for $t \in [u, T-1]$. The obtained, (so called effective reproduction number) R_e with a lower and upper bound as well as $\widehat{R}_e$ is written into the *dan* table.

Step 3. The statistics of R_e and $\widehat{R}_e$ as mean values, standard deviations, standard deviations and the Shapiro test p-value $R_e - \widehat{R}_e$ are computed and written into the *df* table.

Step 4. Treating the values R_e or $\widehat{R}_e$ not as time series but as statistics we may compare the epidemic in other countries. In this aim, we use a function in language R *ks.dist*. It is the Kolmogorov distance — the maximal distance between the cumulated spectra. This distance is a basis for calculating a hierarchical classification of epidemic runs in other countries — command *hclust* (cf. Figure 4).

Step 5. We analyze factors that have an impact on the intensity of epidemics in other countries. As an input, we use stringency index, population density, number of doctors per 100000 residents and climate. The strength of the epidemic is measured by the mean R_e (or $\widehat{R}_e$) value and excess mortality (the outputs). We build view models: linear regression, regression trees and neural nets (*neuralnet, nnet, mlp, jordan, rbf, elman* from libraries *neuralnet* and *RSNNS*). The obtained models are analysed from a viwepoint of minimal residuals. We note that for neural nets analysis, the inputs and outputs should be normalized to the $[0, 1]$ interval, for comparison purposes we make all computations on a normalised data frame. The best model (proved regression tree) was presented in Figures 3 and 4, and the influence of input factors were analysed by the *DALEX* library and shown in Figure 5.

5 Results

5.1 Phases of the epidemic in Poland

Figure 1 shows the optimal computed values of β and γ for the SARS CoV 2 epidemic in Poland for the data that were truncated to times $[u, u+t], t = 5, 6, 7, \ldots$. For Poland, we get $u = 49$, since the first recover/death observation arises after 49 days. Thus, the first β and γ values illustrate the situation of the initial phase of the epidemic. First, for $t \in [5, 225]$ there are two stationary points of (β, γ) : $(1, 0.946)$ and $(0.524, 0.475)$. These points characterize very short times of new transmissions and a short time recovery. These values suggest an overlap of two different epidemic processes (probably in different regions of the country). For the time interval $[227, 270]$ all resolutions were not stable (*ode* procedure shows *Abnormal termination*). This chaotic behavior ends two stationary points in the times $[270, 404] \cup [461, 507]$ with $(0.36, 0.31)$ and in the times $[405, 460] \cup [508, 568]$ with $(0.27, 0.23)$. The process, in these times, is characterised by a longer time of new transmissions and a long time of recovery. The observed changes can be an effect of virus mutations, the immunisation of the population, self-restrictions of social contacts as well as the inaccuracies of available observations.

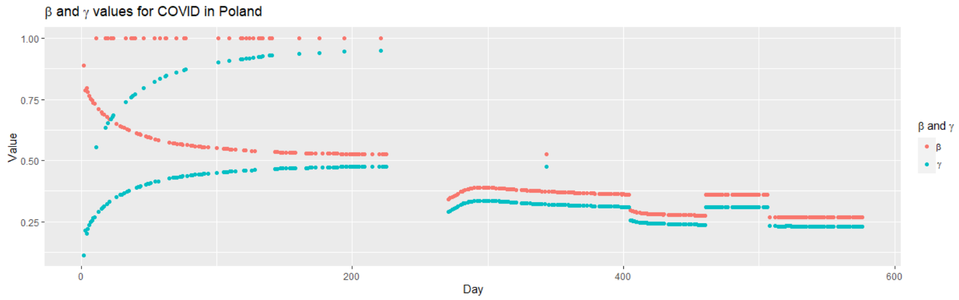


Figure 1: Change in optimal, for the model SIR, γ and β values versus simulation time interval

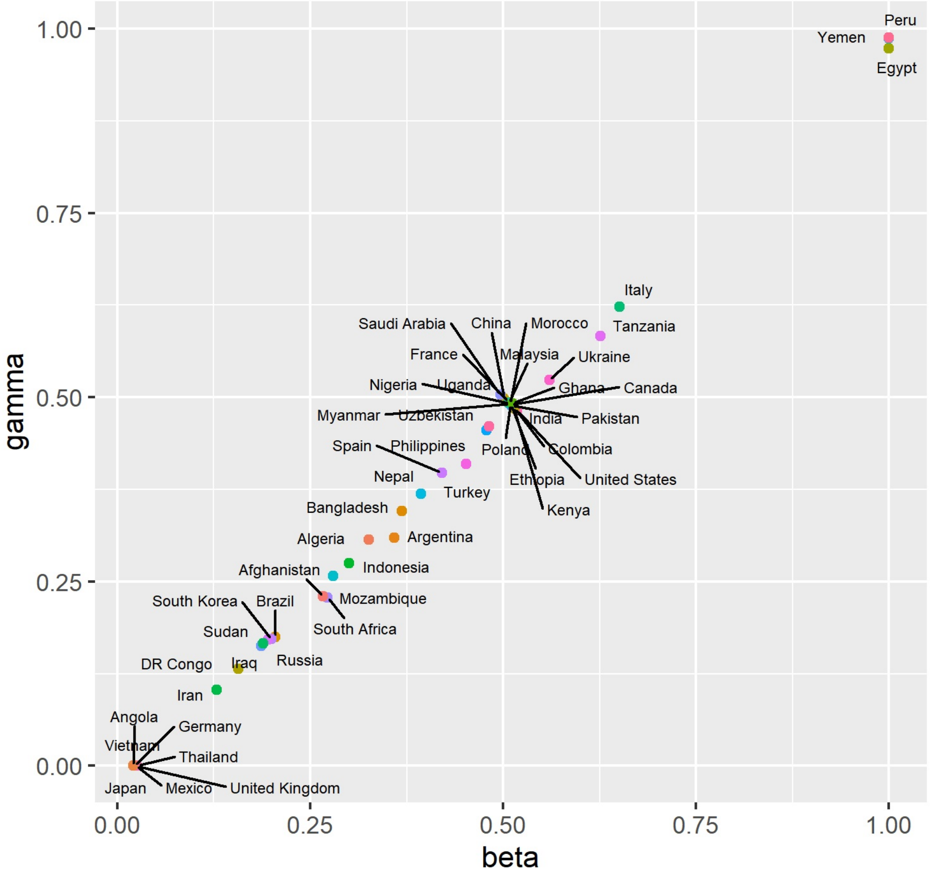


Figure 2: Optimal values β and γ in different countries.

5.2 "Shape" of the epidemic

The "shape" of the epidemic was measured by optimal values of β and γ computed from the maximal times of observation. From Figure 2 we see, that the shape of epidemic in Poland was similar to that in countries such as France, United States, India and China and that it "essentially" differs from that in Germany, United Kingtom, Italy or Ukraine. All values β and γ lie close to the line $\mu \sim \lambda$, thus the R_o value is similar for all countries, but the contagiousness is differ.

5.3 "Intensity" of the epidemic

"Intensity" of epidemic we measure by the mean of (smoothing) R_e or (unsmoothing) $\widehat{R}_e$ values, by the relative (to population size) COVID deaths and the relative of number of excess deaths. Obviously the higher mentioned above values indicate on the stronger running of epidemic. In this and the next subsection we will be treatise the time series values $\widehat{R}_e(t)$ and $R_e(t)$ for $t \in [u, T-1]$ as the statistics. Because from (3) $\widehat{R}_e$ is quotient of two values, thus the median or geometric mean is better, from the interpretations "viewpoint", for evaluations these statistics rather than usual mean. The geometric mean of $\widehat{R}_e(t)$ can be interpreted as:

$$\exp\{\sum_{t=u+1}^{T-1} \ln(\widehat{I}(t+1) - \widehat{I}(t))/(T-u-2) - \ln(\widehat{I}(t)/(T-u-2)\}$$

i.e. the exponent of difference of mean natural logarithms of increments and mean natural logarithms of infected numbers. For the dispersion measure we use the standard deviation of $\ln \widehat{R}_e(t)$ and $\ln R_e(t)$ for $t \in [u, T-1]$, deleting the moments when $\widehat{R}_e(t) = 0$ or $R_e(t) = 0$. These values are given, for countries, in the Table 1. The fragment of densities of $\widehat{R}_e$ values for five countries (Poland, Germany, China, Russia and United States) are presented in Figure 3. For comparison purposes, between different countries, we compute a useful distance measure — the Kolmogorov-Smirnov distance defined for two arbitrary distribution functions F and G:

$$d(F,G) = \sup_x |F(x) - G(x)|,$$

and on this basis we build a hierarchical classification using the R instruction *hclust* (results sre shown in Figure 4).

The results for smoothing R_e are well for the Poland but the unsmoothed $\widehat{R}_e$ are worst, for COVID deaths and especially excess deaths definitely worst. The standard deviation value indicate the stable runs of epidemic.

The obtained $\widehat{R}_e$ values indicate that the run of the epidemic in Poland was worse that Germany, China (now it has probably changed) and Russia and better than that in the United States.

Figure 4 shows that the epidemic in Poland was very similar to that in Tanzania and a little less similar to that in Chile, Hungary, Portugal, Russia and Columbia. It is interesting, the runs of epidemic in Poland's neighbouring country, Germany, are significantly different.

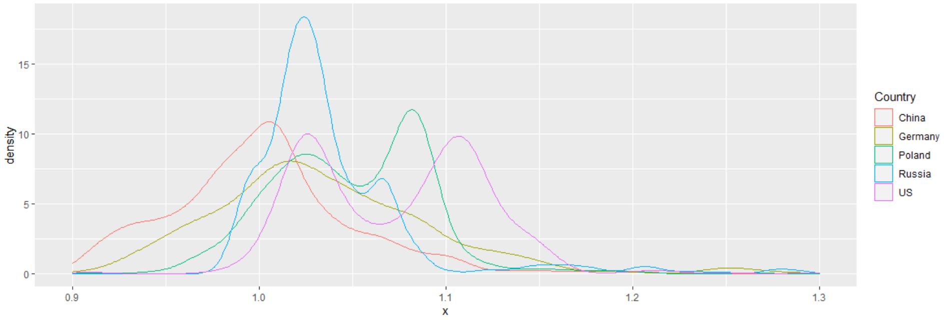


Figure 3: Densites of $\widehat{R}_e$ of the SARS CoV 2 epidemic

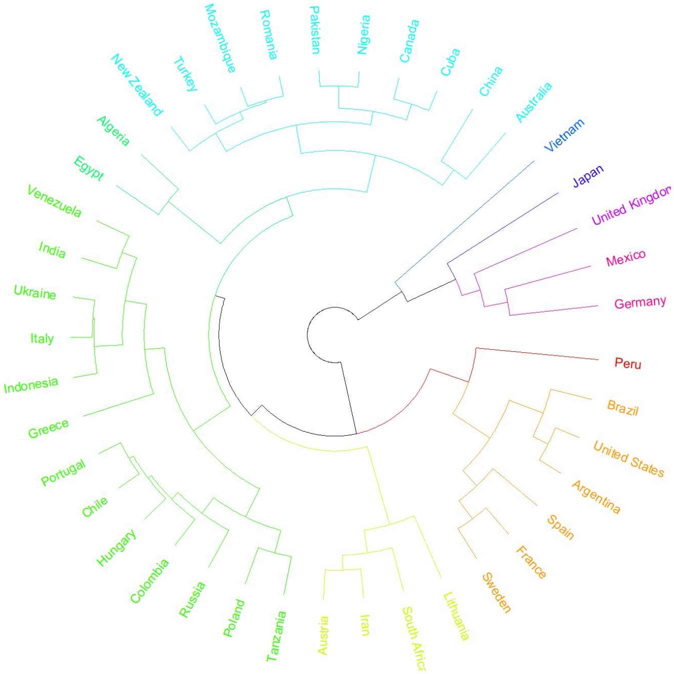


Figure 4: Comparison of R_e distributions based on the Kolmogorov-Smirnov distance in different countries

Table 1: Evaluation of the SARS CoV 2 epidemic in selected countries

Country	Population	R_e		$\widehat{R}_e$		Deaths[d]	Excess
	(N[a])	mean[b]	sd[c]	mean[b]	sd[c]		deaths[d]
United States	334.8	1.254	0.32	1.086	0.09	128.77	148.60
Lithuania	2.7	1.223	0.28	1.071	0.11	147.30	304.89
Brazil	215.4	1.192	0.26	1.07	0.10	136.32	158.13
Israel	8.9	1.154	0.48	1.079	0.10	52.22	50.48
Iran	86	1.146	0.30	1.039	0.07	73.45	139.39
Hungary	9.6	1.141	0.31	1.056	0.06	208.21	176.97
Japan	125.6	1.135	0.37	1.024	0.08	8.71	4.20
Sweden	10.2	1.116	0.14	1.073	0.08	79.56	47.31
Germany	83.9	1.116	0.43	1.04	0.12	70.32	49.47
France	65.6	1.109	0.11	1.068	0.13	93.69	61.53
Netherlands	17.2	1.105	0.16	1.076	0.10	55.05	77.05
Romania	19	1.094	0.18	1.051	0.11	152.53	282.71
Russia	145.8	1.088	0.13	1.045	0.09	112.67	371.04
India	1406.6	1.086	0.12	1.035	0.12	–	–
Chile	19.3	1.085	0.10	1.058	0.10	127.35	110.44
Poland	37.7	1.082	0.15	1.058	0.09	134.74	202.96
Belarus	9.4	1.082	0.13	1.043	0.13	19.11	275.00
Italy	60.3	1.073	0.13	1.055	0.11	118.69	141.28
Ukraine	43.2	1.069	0.11	1.047	0.08	111.48	203.80
Australia	26.1	1.045	0.11	1.017	0.09	6.68	4.80
China	1448.5	1.006	0.12	1.003	0.07	–	–

[a] In milions, [b] Geometric mean, [c] Standard deviation of logarithmed values, [d] Per 100000 persons.

5.4 Factors affecting the shape and the intensity of epidemic

As it was mentioned in the previous section, we measure (as output) the strength of the epidemic by: geometric mean or median of $\widehat{R}_e(t)$ and $R_e(t)$, relative deaths and relative excess death. As an input we take population density, the number of doctors per 100000 people, COVID stringency index [13] and climatic zone (coded as three boolean values *climate_moderate, climate_subtropical* and *climate_tropical*). Stringency index (cf. [6], [13]) is calculated based on the following thirteen metrics: school closures; workplace closures; cancellation of public events; restrictions on public gatherings; closures of public transport; stay-at-home requirements; public information campaigns; restrictions on internal movements; international travel controls; testing policy; extent of contact tracing; face coverings; and vaccine policy. Between the building by us models; regression, decision trees (*rpart*) and neural nets (*neuralnet, nnet, mlp, rbf, jordan, elman*) the best turned out the *rpart* (geometric mean R_e, median $\widehat{R}_e$ and excess deaths) and *neuralnet* (geometric mean $\widehat{R}_e$, median R_e and COVID deaths). The produced best models were investigated in the library *DALEX* to find the input factors that had the greatest influence on the output. Results are given in Figure 5.

It can be seen that the first two positions plays climate value. Therefore, by dividing the countries according to their climate zone, we compute the Person correlation values, which are given in Table 2.

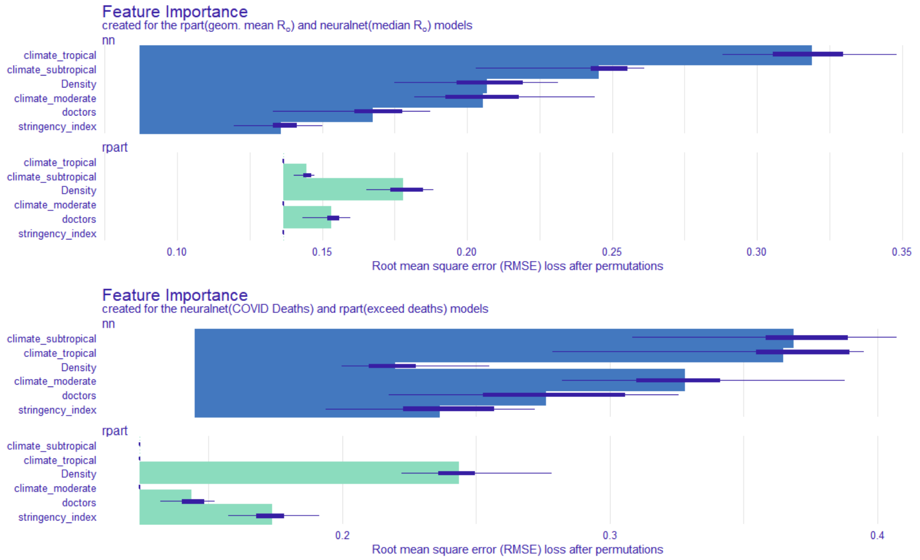


Figure 5: Validation of factors affecting the intensity of epidemic

Table 2: Pearson correlations of factors

Value	Strigency index	Doctors	Density
	moderate climate		
geom. mean R_e	-0.2902	0.3481	0.0716
geom. mean $\widehat{R}_e$	-0.1085	0.3884	0.4983
median R_e	-0.3209	0.3645	0.0541
median $\widehat{R}_e$	-0.1823	0.3179	0.4551
COVID death	-0.0807	0.3920	0.0243
Excess death	-0.2377	0.0698	-0.2604
	subtropical climate		
geom. mean R_e	0.1636	0.1076	-0.2101
geom. mean $\widehat{R}_e$	0.5041	0.3586	0.2405
median R_e	-0.0474	0.1006	0.1296
median $\widehat{R}_e$	0.4480	0.4005	0.2378
COVID death	0.3202	0.3782	-0.0720
Excess death	0.0909	-0.2667	-0.2181
	tropical climate		
geom. mean R_e	-0.0270	0.1010	-0.2153
geom. mean $\widehat{R}_e$	0.3500	-0.0083	-0.2264
median R_e	-0.0004	0.2405	-0.1448
median $\widehat{R}_e$	0.5841	-0.0771	-0.2364
COVID death	0.3977	-0.2161	-0.2343
Excess death	0.1198	-0.2858	-0.2271

6 Conclusions

1. One can distinguish three stages of the SARS-CoV-2 epidemic in Poland: 1–250, 250–450, 450–600 (days).

2. The run of the epidemic in Poland was rather similar to that in South American or African countries rather than that in Europe. As for In European countries, the most similar runs were observed in Hungary, Portugal and Russia. The runs of the epidemic in Poland's neighbouring countries: Germany, Czech and Slovak were considerably different.

3. In a tropical climate, the number of doctors is inversely proportional to the strength of the epidemic (curability is increasing), whereas, for the countries in a moderate or subtropical area, this number is directly proportional to the strength of the epidemic (detectability).

4. For countries with a tropical climate, the density is inversely proportional to the strength of the epidemic (detectability), whereas, for countries with moderate or subtropical climates, the density is proportional to the strength of the epidemic (infectivity). Density is always reversely proportional to the number of COVID deaths and excess deaths.

5. The stringency index is directly proportional for the countries with tropical and subtropical climates and inversely proportional for the countries with moderate climates.

Looking at points 3–5 we see that the epidemic in the tropics is substantially different than this epidemic in countries with moderate climates. The differences are considerable, but more detailed conclusions required further investigations.

References

[1] https://raw.githubusercontent.com/RamiKrispin/coronavirus/master/csv/coronavirus.csv

[2] https://data.europa.eu/euodp/en/data/dataset/covid-19-coronavirus-data

[3] https://data.europa.eu/euodp/en/data/dataset?vocab_concepts_eurovoc=http%3A%2F%2Feurovoc.europa.eu%2F838

[4] https://github.com/owid/covid-19-data

[5] https://statsandr.com/blog/covid-19-in-belgium/

[6] https://ourworldindata.org/covid-stringency-index

[7] Allen, L. J. (2010). An introduction to stochastic processes with applications to biology. CRC press. https://sistemas.fciencias.unam.mx/~silo/Cursos/coronavirus/Allen.pdf

[8] Andersson, H. & Britton, T. (2012). Stochastic epidemic models and their statistical analysis (Vol. 151). Springer Science & Business Media. http://archive.schools.cimpa.info/archivesecoles/20160119113752/t_britton.pdf

[9] Aragon, T. J., Enanoria, W. T., Fay, M., Porco, T. & Samuel, M. (2007). Applied epidemiology using R. Berkeley, Calafornia: MedEpi Publishing.

[10] Arroyo-Marioli, F., Bullano, F., Kucinskas, S. & Rondón-Moreno, C. (2021). Tracking R of COVID-19: A new real-time estimation using the Kalman filter. PloS one, 16(1), e0244474. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0244474

[11] Harko T., Lobo F. S., Mak M. K. Exact analytical solutions of the Susceptible-Infected-Recovered (SIR) epidemic model and of the SIR model with equal death and birth rates. *Applied Mathematics and Computation.* (2014) 236: 184–194.

[12] Kermack W. O. and McKendrick A. G., Contributions to the mathematical theory of epidemics–I., 1927, *Bulletin of mathematical biology*, 53, 1–2, p. 33–55, 1991.

[13] Ritchie H., Mathieu E., Rodés-Guirao L., Appel C., Giattino Ch., Ortiz-Ospina E., Hasell J., Macdonald B., Beltekian D. and Roser M., (2020) — "Coronavirus Pandemic (COVID-19)". Published online at OurWorldInData.org. Retrieved from: https://ourworldindata.org/coronavirus [Online Resource]

Augmentation is not Enough. On Advancements in Self-Supervised Representation Learning for Computer Vision Tasks

Michał Chromiak*

1 Introduction

If we state that — *the measure of intelligence is the ability to change* — we could define intelligence as the efficiency with which one transforms unseen data into new skills. The *"change"* term requires clarification, as the direction of the change needs to be reasonable and adapt to adequate variation of inputs from the surrounding environment. In other words, *intelligence is a process of adaptation, improvisation and generalization to a new environment, while the measure of intelligence is how effective this process is*. In this context, self-supervised learning is about gaining new skills — adapting — based purely on new (i.e. non-labeled) data.

The human brain learns mainly by observation. We infer our understanding of real-world objects by generalizing from smaller experiments to more general physical laws and behaviours. Therefore, tries and errors are the most powerful tool for our cognition that helps us to build hypotheses about how the world works. As it would not be possible to label every object in the world, humans mostly do not need prior knowledge about real-world objects, in form of labels, to learn their conceptual physical representation. In pursuit of achieving increasingly general AI, imitating such an approach would bring a considerable advantage. One of the most promising tools for that are the *foundation models*[1] [2], that introduce scale and the ability to perform tasks beyond training. It means that the system behaviour is implicitly induced rather than explicitly constructed. This way, the same model can be used for a wide range of tasks. The foundation models are powerful transfer

*Corresponding author — `michal.chromiak@mail.umcs.pl`

[1] The foundation model is a model trained at a broad scale that can be adapted to a wide range of downstream tasks. Eg. BERT [13], GPT-2/3 [32, 3], CLIP [30]. Based mostly on neural networks and self-supervised learning (SSL).

learners. Their dominant paradigm is to train a model on a surrogate task to learn good representations and then perform adaptation by performing fine-tuning on the task of interest. The scale differentiator of foundation models is possible due to the computer hardware improvements, availability of massive training data in recent years, as well as introduction of the Transformers [39], which allowed to leverage parallelism and provide expressivity. The initial success of transfer learning was based on supervised pre-training using labelled datasets like ImageNet [34]. However, the high costs of labelling limit the scale-up ability of this approach.

Supervised pretraining, as the most straightforward and thus, a popular method of supervised representation learning becomes insufficient when it comes to big data scales. It requires the acquisition of carefully labelled data, which is time and cost-consuming. With increasing amounts of data, it is almost impossible to manually label at the big data scale. This way supervised learning becomes a bottleneck for scaling solutions that would be used for more general models. In conclusion, supervised methods are not sufficient for more general tasks thus, the interest should focus on non-label-dependent, unsupervised approaches.

One of the most promising approaches to tackle the problem of massive amounts of unlabeled data is *self-supervised learning*(SSL). The SSL is similar to unsupervised learning[2], as it is not based on any labels. However, in the case of SSL, the goal is to solve tasks that are traditionally solved by supervised learning despite the lack of labels. SSL aims at solving the problem of learning from massive amounts of unannotated/unlabeled data. Therefore, we can state that SSL generates labels out of the data itself. This way SSL is inherently more scalable than supervised algorithms since they have no associated labelling cost. Being based on predictions about missing/unobserved/hidden parts of (potentially very diverse) input from its observed parts enables SSL to learn potentially a richer learning signal, as it is based on context, rather than more limiting, finite label space.

Historically, self-supervision research has been present in the area of natural language processing (NLP), in different flavours. Initially, the idea has been used for *word embeddings* that could be learned to associate words with context-independent vectors (word representations [37], word2vec [26], or GloVe [27]), which later could be used to downstream tasks. This approach evolved to autoregressive language modelling as a pretraining stage to produce context-aware representations (seq2seq pretraining [11], GPT [31], ELMo [28] or ULMFit [20]). Eventually, self-supervision has been coupled with Transformers [38] resulting in ideas such as BERT [13], GPT-2 [29], RoBERTa [25], T5 [33] or BART [22].

As for the recent AI breakthroughs, they still originate mostly from two areas of research. It is the SSL, which helps to find dependencies from unseen data, and the Transformers, which allows a model to act more reasonably by selecting only important parts of the input data to focus on. Moreover, since BERT (2019), self-supervised language models (LMs) have become a substrate of NLP. The usage of foundation models (used as a single model for multiple tasks) has become the norm nowadays. This is referred to as *homogenization*, where instead of hand-crafted features for each task, the same architecture is being used for multiple tasks. This trend also holds for other areas of research where Transformers finds

[2] Unsupervised learning is more focused on finding high-level patterns for clustering or dimensionality reduction.

their applications such as computer vision (ViT [14]), speech (Mockingjay [24]), tabular data (ToBERT [40]), or reinforcement learning (Decision Transformer [6]).

Out of the two discussed research areas, the Transformers has already gained impressive results in computer vision AI research [9]. However, SSL recently is also gaining more attention, especially in computer vision where it is crucial to have an efficient way of representing each image in a latent space. Such representation can be learned. In the most general case, this means learning a neural network (like ResNet-50) to represent an image as a vector, without access to ground-truth annotations. In this paper, we will focus on recent SSL advancements in representation learning for computer vision tasks.

2 Challenge of SSL for computer vision

Supervised representation learning is based on training some modality (e.g. an image) representation based on a big, labelled dataset like the ImageNet [12]. Such learned representation can later be transferred to a task that might not have enough data to train good latent representation. The next step is simply to fine-tune general image representation to a specific task. This method originated from natural language processing (NLP).

The solutions based on big transformers, like BERT, RoBERTa, XLM-R [10] and others, encodes the common representation from general, large, labeled dataset and then adapt it to nuances of the target dataset of downstream tasks.

However, in the area of computer vision (CV), SSL tends to be more challenging than in the case of NLP. The reason for this is that prediction of, say a missing word in a sentence, requires the computation of a probability score with some kind of a *softmax* layer for every possible word in a vocabulary that is large, but countable and finite[3]. Ultimately it is being shaped as a classification task of a discrete domain.

2.1 Uncertainty and continuous domain

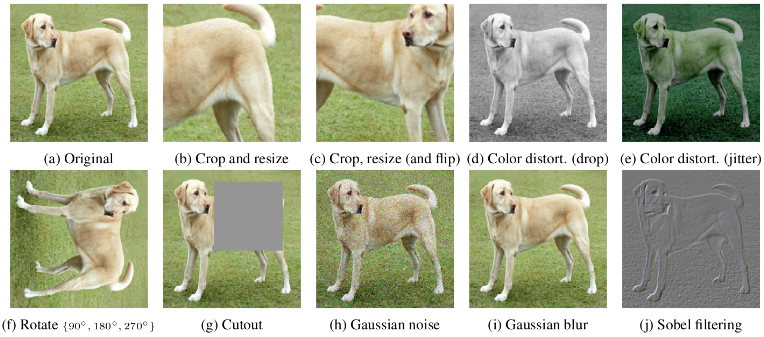
In contrast to NLP, the CV or speech prediction tasks focus on high-dimensional continuous objects rather than discrete words. It means that analogous tasks in CV need to consider the prediction of a token that would be a missing part of an image or a video frame. Similarly, in speech, it would mean finding the missing segment of a recording. In those domains, the tokens are not limited to countable vocabulary but spread across a continuous space of solutions resulting in an infinite number of potential outputs.

In the image domain, the challenge is also to represent the uncertainty within this continuous space of solutions. For NLP tasks, a sufficient solution would be to apply a *softmax* function over the finite number of possible outputs. This also allows to an expression of uncertainty in NLP, simply by outputting discrete probability distribution across possible labels. However, in case of the continuous, computer vision tasks — the challenge is more demanding as it requires finding the right predictions from high-dimensional solution space in the presence of uncertainty.

[3] Vocabulary size is a matter of choice.

2.2 Lack of labelled images

The lack of labelled examples for training is what defines SSL. In recent research one of the most popular methods of dealing with the lack of labels in computer vision uses image augmentation. This concept keeps making multiple variants of the input image by changing the picture to some extent so that it still contains sufficient information to be treated/represented as the original, input image. The goal is to learn a neural network to output a latent representation of an image that would produce similar embeddings for the same image variants. Modifications applied to the original image include operations such as random cropping, colour distortions, rotations, flipping, resizing, Gaussian blurring, etc.



(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

(f) Rotate $\{90^\circ, 180^\circ, 270^\circ\}$ (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

Figure 1: Examples of data augmentation operators used in contrastive learning [7]

Having the unmodified image and a set of its modified variants (Fig. 1) one can train a neural network (usually some ResNet-50 variant) model to learn representation for the unannotated image.[4] This technique of learning representations with objective functions where inputs and annotations are derived from unlabeled data is recently being heavily under research due to achieving state-of-the-art results.

3 Learning visual representations

In self-supervised image representation learning, the most common approach was to use some kind of ResNet (eg ResNet-50) neural network to learn function capable of representing image as a vector. Such learned representation of an image needs to be useful later in transfer learning. It would be used for fine-tuning in multiple downstream tasks with potentially smaller image training sets. In contrast to supervised version, the SSL solutions do not have labeled data in the initial representation learning step. The challenge of lacking labels however can be solved with use of original image *augmentations*.

The long-standing problem of learning representations without human supervision (i.e. from unlabeled data) is solved mainly by two classes of approaches, namely: *generative* and *discriminative*.

[4] For example to train a linear classifier on top of those representations of unlabeled data.

The generative approach [15] in vision is based on modeling pixels in the input space and thus, comes at computational costs. On the other hand, the discriminative approach (similarly to supervised learning) use objective function to work, on unlabeled data. Recently, the discriminative *contrastive learning* [1, 7, 8] approach has gained lots of attention due to achieving better than state-of-the-art results. One representative architecture that benefits greatly form contrastive learning is SimCLRv2[5] [8] that was able to surpass even supervised alternatives by dint of utilizing augmentations.

3.1 Contrastive Learning

With *contrastive learning* representations are discovered by maximizing similarity between differently augmented variants of the same (original, aka *anchor*) datapoint via a contrastive loss in the latent space. Contrastive loss is a function that actually makes the anchor's variants (aka *positive samples*) close to anchor, and all of the remaining, i.e. *non-anchor* related (aka *negative samples*) images, to be far away. Obviously this rise question for classification problems, namely: how to make not only augmentations of anchor to be close in latent space, but also other instances of the same class with their augmentations to be close as well? The methods shown in [21] prove that this can be done with partial use of labels — i.e. supervised learning. This *supervised contrastive learning* approach states that creation of decision boundry can be done during first stage with self-supervised pretraining. Next, in stage two, when the representations are already learned (and datapoints of specific classes are separated), a classifier is trained with supervised cross-entropy loss with softmax. Specifically, with SimCLRv2 [8], first stage is to learn general representation without labels in a task-agnostic way, while the second step uses small number of labeled examples to learn specific task with a supervised fine-tuning.

Another approach discussed in [21] is based on, so called *Triplet Loss*. It requires one positive, and one negative sample for each anchor. However, the downside of this solution is that having only one negative sample requires this sample to be *hard negative* to deliver enough learning signal for a triplet case. This brings us to a general challenge of finding good negative samples for each case, which significantly complicates the process.

In SimCLR, the problem of choosing optimal positive and negative samples is addressed by the augmentation scheme. It is possible via generation of two augmentations[6]from each anchor (see Fig. 2). Such pair is evaluated with contrastive loss function (see Eq. 1) as "close" (numerator of loss function; two augmentations of same image) to each other, while all the remaining anchors and their augmentations are to be moved away (denominator of loss function) in representation space. As SimCLR framework tends to be important baseline for this research area, therefore let's formalize its stages:

[5] Not fully self-supervised, as it uses semi-supervised approach however, it uses significantly less labels than fully supervised solutions. With only 1% of labels, SimCLRv2 is on pair with Top-1 accuracy of supervised ResNet-50 — see [8] Figure 2.

[6] One obvious drawback of this idea is that uses random sampling for different classes from batch, Therefore, when applied to imbalanced dataset (SimCLR authors used perfectly balanced dataset — ImageNet), it would result in drop of performance.

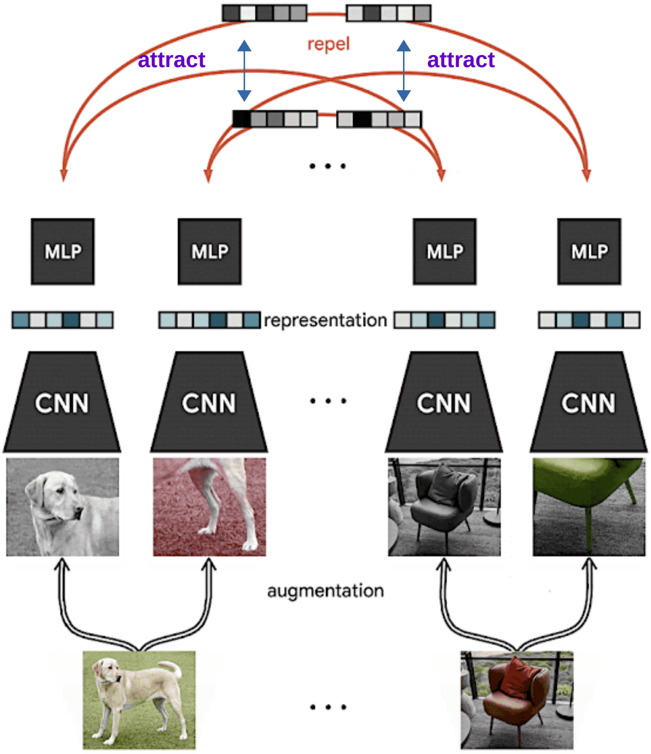


Figure 2: *SimCLR* contrastive architecture illustration

1. *Positive sampling module*: A stochastic data augmentation is done by sampling two augmentations from the same family of augmentations $t, t' \sim \mathcal{T}$ of original datapoint x: $\tilde{x}_i = t(x), \tilde{x}_j = t'(x)$. This way two correlated views are created.

2. *Base encoder*: Before an augmentation ever becomes an input of a contrastive loss, it is must be encoded via encoder network $f(\cdot)$[7]to create embeddings.

3. *Projection head*: Transforming representation vectors with non-linear transformation network $g(\cdot)$ (MLP projection head) that compress representation to lower the dimension for inner product of contrastive loss: $z_i = g(h_i) = W^{(2)}\sigma(W^{(1)}h_i)$, where σ is a ReLu non-linearity.[8]

4. *Contrastive loss function*: For a given set of $\{\tilde{x}_k\}$ (including the postive pair $\{\tilde{x}_i\}$, $\{\tilde{x}_j\}$), the contrastive prediction task for a given $\{\tilde{x}_i\}$ identifies $\{\tilde{x}_j\}$ in $\{\tilde{x}_i\}_{k \neq i}$

[7] The choice of encoder function can be arbitrary, but in most cases (like with SimCLR family) the encoder is just a ResNet [18].

[8] Authors claim that leveraging the nonlinear projection g (FC→ReLu→FC→Loss) helps to maintain more information in h.

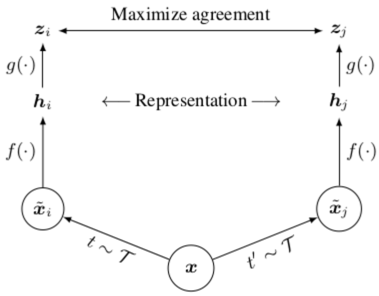


Figure 3: *SimCLR* framework [7]

The randomly sampled N inputs from minibatch are then augmented to $2N$ views. For each positive pair all of the remaining $2(N-1)$ augmented samples are treated as negatives. With cosine similarity[9]- *sim* operator — being dot product of ℓ_2 normalized arguments

$$l_{i,j} = -log \frac{exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp\left(sim(z_i, z_k)/\tau\right)} \tag{1}$$

Used loss is NT-Xent (the normalized temperature-scaled cross entropy loss) with τ denoting temperature[10]

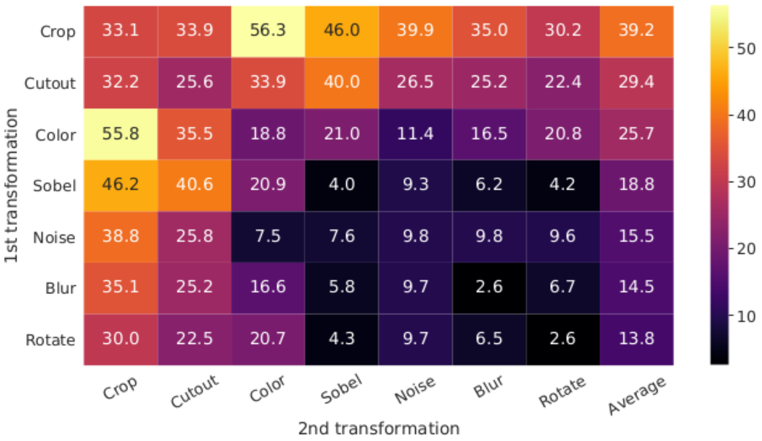
It brings us to point where for positive samples we minimize the loss, and for negative samples we maximize the contrastive loss. Therefore, the whole process boils down to simply training model to represent augmentations of the same image close, within latent representation, and away for different images via contrastive loss. It means that the models actually learns to *ignore* augmentations used in the training stage, while deciding about similarity of two given datapoints during inference.

3.2 Augmentations

One important finding form SimCLR is how meaningful it is to pick the right augmentation scheme. It turns out that composing augmentations has significant positive impact on the representation quality. It is especially visible in case of random cropping and random color distortion. It sounds natural due to the fact that cropped-only image patches share similar color distribution as confirmed via color histogram analysis. This way it is important what kind of augmentation scheme will be picked in the framework.

[9] $sim(x, y) = x^T v/\|x\|\|y\|$

[10] The temperature scaling is used to *calibrate neural networks* by raising the output entropy without impacting model's accuracy. Temperature scaling softens outputs of the neural network. Therefore network becomes less confident. As a result it makes the confidence scores reflect true probabilities.



1st transformation \ 2nd transformation	Crop	Cutout	Color	Sobel	Noise	Blur	Rotate	Average
Crop	33.1	33.9	56.3	46.0	39.9	35.0	30.2	39.2
Cutout	32.2	25.6	33.9	40.0	26.5	25.2	22.4	29.4
Color	55.8	35.5	18.8	21.0	11.4	16.5	20.8	25.7
Sobel	46.2	40.6	20.9	4.0	9.3	6.2	4.2	18.8
Noise	38.8	25.8	7.5	7.6	9.8	9.8	9.6	15.5
Blur	35.1	25.2	16.6	5.8	9.7	2.6	6.7	14.5
Rotate	30.0	22.5	20.7	4.3	9.7	6.5	2.6	13.8

Figure 4: *SimCLR* Top-1 accuracy for composition of two data augmentations

3.3 Distillation and self-training

SimCLRv2 [8], apart from larger ResNet and deeper projection head, also utilizes semi-supervised phase to fine-tune with labels as the second stage. Moreover, SimCLRv2 has embraced additional technique called *distillation*. Distillation becomes the third stage of SimCLRv2. It uses the fine-tuned network as a *teacher* to input labels for training a *student* network. Teacher is trained with first two stages: unsupervised pretraining and supervised fine-tuning. The considered cases involved: the student network that shares the model architecture with the teacher[11], and the distillation where student is smaller network, than the teacher network.

Improvement of the distillation with unlabeled datapoints in SimClRv2 is twofold. Depending on the relative size of student it — improves model efficiency by transferring task-specific information to a student model — in asymmetric case (i.e. student has smaller architecture than the teacher model), or improves semi-supervised learning performance in the symmetric case (same architecture of student and teacher networks).

In the first case, teacher (large, trained) model infers about an input generating some pseudo label. Then, loss function compares output of the (smaller) student, for the same image, with pseudo-label from the teacher network. It is the *self-trained* example using the unlabeled datapoint.

The second case, is where the teacher and student networks are the same (self-distillation) and in that case the student model, with use of unlabeled examples and the freezed teacher, can get better performance than the teacher itself.

This way SimCLRv2 demonstrates, that with use of unlabeled examples, a general representations can be distilled into a more specialized, compact student network. It can even beat the supervised learning with only 1% of its labels.

[11] Except form projection head.

4 Reducing the need for negative samples in self-supervised learning

As already mentioned in 3.1 the key to SSL is learning a general representation from unlabeled dataset with use of contrastive methods. They reduce distance (in latent space; according to loss) between representations of same image augmentations — *positive samples* — and increasing the distance to augmented views of different anchor images — *negative samples*. The key challenges of those methods were: how to choose negative pairs (so that they can provide significant learning signal[12]), and which augmentation compositions to use. There are many questions about how to choose negative samples, like: shall they be uniformly sampled, should they be buffered output of memory network [17], shall one use hard negative mining[13]or semi-hard[14], would they be ordered — curriculum learning [35] challenge.

Despite such complexity, the requirement for negative samples has been motivated by conjecture that having only positive samples will make the network to degenerate. Which means that representations of the image augmentations would simply collapse to some trivial, constant solution, providing no value. Negative samples were expected to prevent such convergence to same vector for all images.

One interesting solution has proven not to be constrained by its contrastive counterparts, as it has eliminated the need for negative sampling, while staying less prone to the choice of augmentations. This self-supervised approach is called *Bootstrap Your Own Latent* (BYOL) [16].

BYOL uses two networks: *online* and *target* to work on two random augmentations of input image. Goal of the architecture is to learn representation that would ignore both augmentations and all the non-semantic information from parameters of the online and target networks.

The architecture first step is to make two random augmentations $t \sim \mathcal{T}, t' \sim \mathcal{T}$ of the same image x. Each augmentation view $(v = t(x), v' = t'(x))$ becomes input for online and target networks. Those augmentation need to be learned by the network to be ignored. BYOL uses the given, *target* representation to train new, potentially enhanced *online* one. As this process is iterative, in each iteration online network becomes the subsequent target network for training new, improved online representation[15]. What is more, BYOL refines the new representation of target network in each iteration with slowly moving exponential average of the past online network.

Both networks have the same architecture, and the difference is in their weights: online θ, target ξ. The online architecture comprises of three stages: an *encoder* f_θ — producing representation y_θ, a *projector* g_θ — producing projection z_θ, and

[12] Negative samples need to be sufficiently close to the positive samples, so that the discriminative approach can be challenging enough to learn reliable representation.

[13] Where the positive sample of an anchor is less similar to anchor than the negative sample from a different class. Thus, cases when the distance metric in form of loss function fails to capture semantic similarity.

[14] Within the loss margin and closest in distance, but farther than positive examples.

[15] This is where the name comes from. The bootstrapped outputs of one network are used as a targets for better representation.

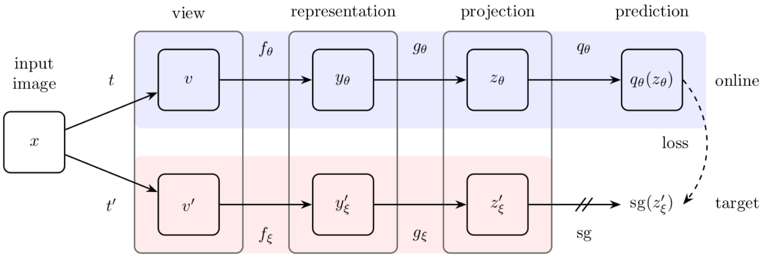


Figure 5: BYOL: architecture

a *predictor* q_θ resulting with prediction $q_\theta(z_\theta)$.[16] θs are trained while ξs are exponentially moving average of the online θs (inspired by RL [23]) to assure smoother changes in the target representation. Each update step considers decay rate τ.

$$\xi \leftarrow \tau\xi + (1-\tau)\theta, \quad \tau \in [0,1] \tag{2}$$

The target network itself, provides the regression target to train the online network.

Notably, there is no explicit part of the design that would prevent BYOL to collapse into trivial solution[17] . The initialization and the learning procedure (that is commenced in small steps) itself can be assumed as the key components that prevent collapse and help algorithm to learn a good representation.

From the neural architecture perspective BYOL uses only ResNets and MLPs (Fig. 6).

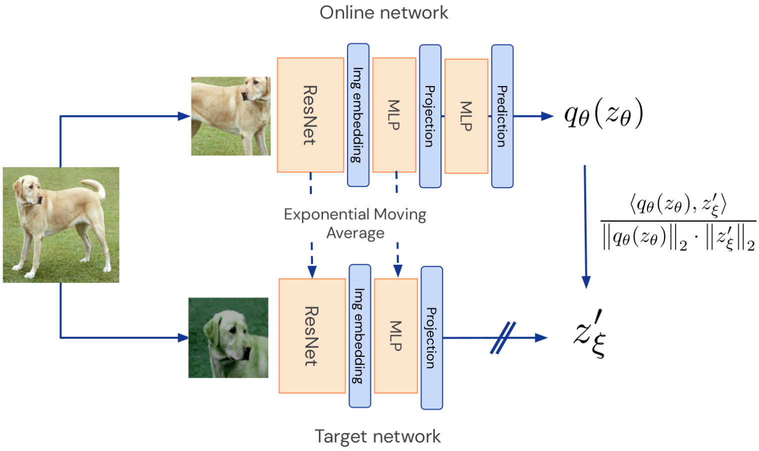


Figure 6: BYOL neural architecture

[16] The predictor is only applied to the online network, making the architecture asymmetric between both pipelines — online and target.

[17] Such as negative samples from the previously discussed solutions.

BYOL is also dependent on augmentations that are specific to vision. For other modalities it is not explored what kind of augmentations would also be similarly suitable for generalizing to other domains.

5 Transformers in SSL

Transformers have already been successfully adopted to multiple computer vision tasks [9].

It turns out that coupling of vision transformers (ViTs [14, 9]) and self-supervision result with architectures that outperforms previous state-of-the-art outcomes. A significantly successful recent attempt to couple SSL with ViT is a framework called DINO.

5.1 Self-distillation with no labels

DINO (*self-**di**stillation* [19] *with **no** labels*) [5] is a methodology for unsupervised pretraining of visual transformers (ViTs) with the ability to learn representation from unlabeled data. Interestingly, the system has not been trained to understand any of the concepts from processed images, nor has it been given a segmentation objective. Nevertheless, its attention maps (Fig. 7), can be easily interpretable as features of higher-level objects. In vision transformer (ViT) the `[CLS]` token embedding (on heads of the last layer) aggregates all the information about an image and is used to do the representation learning. When using `[CLS]` token as a query for heads the result Fig. 7 outcomes with a surprisingly well-segmented image. It means that DINO's attention maps — even without labels, nor supervision — reveal class-specific features which in an unsupervised way lead to image segmentation.

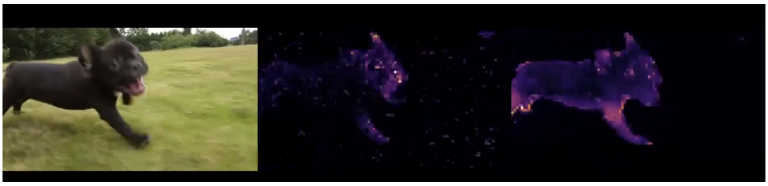


Figure 7: Left: original image, middle: supervised segmentation model output, right: attention map from DINO [5]

DINO can focus on the correct parts of high-level objects and track them even when behind occlusions. DINO's features contain explicit semantic segmentation information of an image, which give it an advantage over supervised ViT, or convnets — which are not that explicit.

Despite sharing the overall architecture with recent SSL approaches (like SimCLR, BYOL, MoCo, etc.), DINO tries to minimize inductive bias by eliminating many of the mechanisms required by these predeceasing solutions. Since DINO is SSL solution, there are no predefined labels, moreover, there are also no *contrastive learning*, nor *negative sampling* mechanisms.

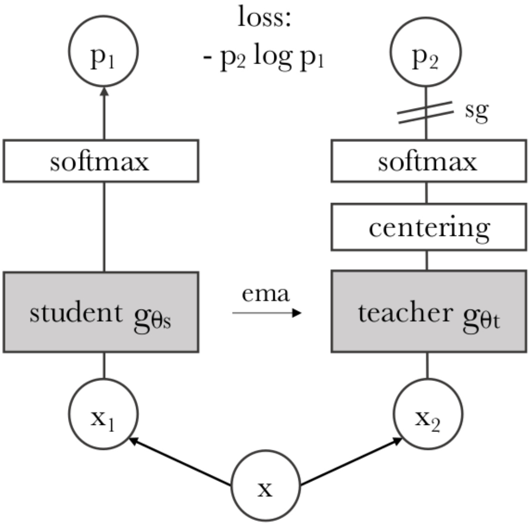


Figure 8: DINO architecture [5]

5.2 Multi-crop, momentum encoder and cross entropy

DINO draws heavily from past research of its authors — SwAV [4] algorithm. DINO is also based on knowledge distillation [19] paradigm that trains *student* network g_{θ_s} to match output of teacher network g_{θ_t} (see Fig. 8).

The inputs for both networks are augmentations of the same image; created with *multi-crop* strategy [4]. These views include two *global* crops (x_1^g, x_2^g) and multiple *local* crops for each image[18]All views are passed through the student network, while only global ones are passed via the teacher network.

Both networks have the same architecture, however, both networks update their weights differently.

The student parameters θ_s are being learned by minimizing the cross-entropy H Eq. 3 with stochastic gradient descent w.r.t θ_s parameters of the student network.

$$\min_{\theta_s} \sum_{x \in x_1^g, x_2^g} \sum_{\substack{x' \in V, \\ x' \neq x}} H(P_t(x), P_s(x')), \tag{3}$$

Additionally, as we want the gradients to propagate only via the student network, the stop-gradient (sg) operator is applied to the teacher.

On the other hand, in contrast to knowledge distillation [19], the teacher network weights are updated with *exponential moving average* (EMA) (teacher network build with momentum encoder [17]) of the student network weights. In turn, this update is similar to what BYOL paper has proposed (in Eq. 2): $\theta \leftarrow \lambda\theta_t + (1 - \lambda\theta_s)$. This way DINO can be treated as a form of *mean teacher* (MT) [36] that is non-semi-supervised *self-**distillation** with **no** labels.*

[18] The global view is >50%, and the local is <50% of the original image size.

Both networks return individual probabilities $P_{\{s,t\}}$ over K dimensional output. P is simply a *softmax* normalization of each network output with additional parameter of temperature τ, that controls the sharpness of the output distribution.

$$P_s(x)^{(i)} = \frac{\exp(g_{\theta_s}(x)^i/\tau_s)}{\sum_{k=1}^{K}\exp(g_{\theta_s}(x)^k/\tau_s)}, \quad \tau > 0 \tag{4}$$

5.3 Centering and sharpening

To avoid collapse earlier solutions used multiple techniques that are no longer required with DINO. Specifically, DINO does not require to use: *predictor* [16] (thus, teacher and student networks have the same symmetric, architecture), *contrastive loss* [17], nor *batch normalization* (BN) [16].

To avoid *collapse* DINO uses *centering* and *sharpening* of the *momentum teacher* outputs. These two operators tend to balance each other as they have the opposite effect. The centring prevents any of the output dimensions to dominate (thus, it favours the collapse of the output to the uniform distribution), while the sharpening encourages domination tendency across the outputs of the teacher.

Centering is based on adding bias c to the teacher: $g_t(x) =\leftarrow g_{t(x)} + c$; with the c being updated with EMA for batch size B and smoothing parameter m:

$$c \leftarrow mc + (1-m)\frac{1}{B}\sum_{i=1}^{B} g_{\theta_t}(x_i), \quad m > 0 \tag{5}$$

The sharpening operator, on the other hand impacts the output and enforces dimension domination by lowering the value of the temperature parameter τ_t for P_t in Eg. 4

6 Conclusions

The SSL algorithms are a very important part of modern research, especially in the computer vision domain. Due to its specificity, this modality is strongly dependent on augmentation schemes.

All of the presented state-of-the-art solutions required a well-designed augmentation to function. However, it's also clear that having the right augmentation is just the first stage of a successful algorithm. The subsequent stages of the presented SSL methods tend to differentiate and improve not so much the augmentation scheme as the knowledge generation mechanics.

The key role of augmentations is to choose distorting characteristics of the image that we think are irrelevant to the representation learning task. This kind of information, makes the algorithm know which image features (colour, brightness, etc.) should be ignored during the training of a representation. Having this in mind, augmentations are the mean of introducing external (non-trained) knowledge into the algorithm. Therefore, the future direction of research that expects a fully autonomous SSL should get rid of this dependency on manual augmentation design.

Another key element in training the SSL algorithm is the data set itself. In the case of an image domain, these may be images that already represent the object of interest, thereby implicitly expressing the dominant information about the image. For example, images of cats, dogs, etc. already help the algorithm by defining a key object to be processed by any algorithm. The information should come from the dataset itself, but the way the dataset is built may already contain indirect indications about representation.

Future research would have to focus around generalizing (or automating) these aspects so that these two inductive biases would not affect the performance of self-supervised learning.

References

[1] Philip Bachman, R Devon Hjelm, and William Buchwalter. *Learning Representations by Maximizing Mutual Information across Views.* Curran Associates Inc., Red Hook, NY, USA, 2019.

[2] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258, 2021.

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020.

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9650–9660, 10 2021.

[6] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *CoRR*, abs/2106.01345, 2021.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 07 2020.

[8] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 22243–22255. Curran Associates, Inc., 2020.

[9] Michał Chromiak. Exploring recent advancements of transformer based architectures in computer vision. In Jarosław Bylina, editor, *Selected Topics in Applied Computer Science*, volume 1, chapter 3, pages 59–75. Maria Curie-Skłodowska University Press, Lublin, 2021.

[10] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116, 2019.

[11] Andrew M. Dai and Quoc V. Le. Semi-supervised sequence learning. *CoRR*, abs/1511.01432, 2015.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021.* OpenReview.net, 2021.

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020.

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, 2020.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[19] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[20] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[21] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *CoRR*, abs/2004.11362, 2020.

[22] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461, 2019.

[23] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *ICLR (Poster)*, 2016.

[24] Andy T. Liu, Shu wen Yang, Po-Han Chi, Po chun Hsu, and Hung yi Lee. Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 05 2020.

[25] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[26] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR*

2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings, 2013.

[27] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[28] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *CoRR*, abs/1802.05365, 2018.

[29] A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021.

[31] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[32] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[33] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683, 2019.

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

[35] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. *CoRR*, abs/2101.10382, 2021.

[36] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[37] Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, page 5998–6008. Curran Associates, Inc., 2017.

[40] Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. Tabert: Pretraining for joint understanding of textual and tabular data. *CoRR*, abs/2005.08314, 2020.

Author index

The second volume of the publication prepared by the Institute of Computer Science of Maria Curie-Skłodowska University in Lublin presents works prepared primarily by young researchers – together with their scientific supervisors working at the Institute.

Chapters will raise very current research problems such as:

- the use of Python with GPU for numerical calculations;
- an experimental study by data science methods of relaxation techniques influence on measurable body parameters;
- a study of the epidemic and pandemic of the SARS-CoV-2 virus – both in terms of social relations as well as the spread of the virus itself – with the use of computer science and mathematical methods.

The whole is topped with a chapter devoted to machine learning used for computer vision – and its future and challenges.

We hope that this small book will be interesting for everyone who is indifferent to what is happening in contemporary computer science – in particular for students and employees associated with broadly understood IT.

UMCS
WYDAWNICTWO

wydawnictwo.umcs.eu

ISBN 978-83-227-9675-7