

Parallel Optimization of Dimensionality Reduction Methods for Disease Prediction: PCA and LDA with Dask-ML

Lesia Mochurad¹, Liliana Mirchuk¹ and Anastasiia Veretilnyk¹

¹ Lviv Polytechnic National University, 12 Bandera street, Lviv, 79013, Ukraine

Abstract

In modern medicine, an urgent problem is to improve the results of cancer prognostication, in particular, to increase the accuracy and reduce the time to obtain a solution. In this paper, we propose to reduce the dimensionality of data at the preprocessing stage using principal component analysis (PCA) and linear discriminant analysis (LDA) methods in order to compare their performance and efficiency. It is known that these methods can be time-consuming, which is critical when solving a forecasting problem. To overcome this problem, the paper proposes to parallelise PCA and LDA methods based on Dask-ML technology. The ResNet-50 model was used to diagnose the disease. The proposed approach has achieved an accuracy of 85.2%, which is 3% higher than the results reported in previous studies. The results obtained indicate that data preprocessing and dimensionality reduction can avoid incorrectly set tasks and improve the accuracy of prediction. In addition, we were able to significantly reduce preprocessing time by parallelising the PCA method using parallel computing technology. In future research, we plan to further improve medical data processing methods, including exploring other approaches to dimensionality reduction and integrating the latest machine learning algorithms to improve prediction accuracy.

Keywords

Medical data processing, machine learning, ResNet-50, parallel computing, cholangiocarcinoma diagnosis ¹

1. Introduction

Timely diagnostics in medicine is extremely important as it allows detecting diseases at early stages, which increases the effectiveness of treatment, reduces its cost and prevents the development of complications. Early diagnostics also improves the quality of life of patients, reduces the burden on the healthcare system, and contributes to the prevention and control of infectious diseases [1]. The development of technologies, such as artificial intelligence, increases the accuracy and speed of diagnostics, making it a key element of modern medicine [2], [3].


It has been reported [4] that patients with cancer had an overall average time to diagnosis of 156.2 (164.9) days, and 15.4% of patients waited longer than 180 days before receiving a diagnosis. Computer diagnostics based on deep learning using images of pathological tissues are often used in cancer diagnosis. However, despite the availability of databases for cancer detection, we still do not have an accurate method for predicting the disease. There are significant difficulties in histological examinations, which are very important for the diagnosis and treatment of diseases. They consist in detecting cancer in tissue images, where scientists often face inverse problems that are incorrectly posed and require special attention to solve [5]. In addition, huge amounts of medical data are generated every day, and their analysis is complicated by factors such as noise, missing data, and high dimensionality. For example, the diagnosis of malignant tumours requires the use of various sources of information [6]. To improve the work with medical data and overcome the difficulties encountered in their analysis, preprocessing is used [7]. Our analysis of the scientific sources has

ProfIT AI 2024: 4th International Workshop of IT-professionals on Artificial Intelligence (ProfIT AI 2024), September 25–27, 2024, Cambridge, MA, USA

✉ lesia.i.mochurad@lpnu.ua (L. Mochurad); liliana.mirchuk.shi.2022@lpnu.ua (L. Mirchuk); Anastasiia.veretilnyk.shi.2022@lpnu.ua (A. Veretilnyk)

ORCID 000-0002-4957-1512 (L. Mochurad); 0009-0000-6772-7552 (L. Mirchuk); 0009-0008-4118-6223 (A. Veretilnyk)

© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

shown that parallelisation of preprocessing can provide better results compared to a sequential process [8].

The relevance of the research conducted in this paper is that there is a lack of efficiency in timely cancer diagnosis, which can lead to severe consequences for the health and life of patients, insufficient accuracy of the results [9] and few solutions for processing multidimensional databases consisting of a large number of images in medicine. In this study, we consider various parallel computing methods and technologies to reduce the multidimensionality of the database at the preprocessing stage, which will allow for better accuracy and not significantly increase the overall solution time.

Challenges in analysing and predicting patient diagnoses include issues such as incomplete or inaccurate data, poorly formulated models, incorrect assumptions, high data dimensionality, and lack of standards or methodological guidance. These factors can make it difficult to make accurate predictions, lead to distorted results, and degrade the quality of diagnosis. Correcting these problems requires correct data preprocessing methods, adequate mathematical models and clear standards to ensure the accuracy and reliability of predictions.

Preprocessing, as a way of solving ill-posed problems, helps to avoid complexities, in particular the 'curse of dimensionality', by reducing the multidimensionality of the database to smaller dimensions that still retain important information. This includes methods such as linear discriminant analysis (LDA) and principal component analysis (PCA), which are used to improve data quality in machine learning, in particular to increase classification and regression accuracy [10]. The optimal choice of preprocessing methods demonstrates significant improvements in prediction, emphasising the importance of this stage in data processing.

In [11], the authors propose a multidimensional choledochal database, which we used to test the effectiveness of the proposed approach. This database contains both microscopic hyperspectral images and colour RGB images in the same field of view for deep learning studies. All the images in this database have been evaluated and labelled by experienced pathologists, making them suitable for training neural networks. This database is very useful for researchers to learn new multivariate deep learning algorithms for pathological diagnosis, as it contains morphology, spectrum, and information about biochemical changes of the samples. Few three-dimensional databases for research have been published on the Internet. To date, the presented multidimensional choledochal database is the first publicly available database of choledochal pathology that contains both microscopic hyperspectral and colour RGB images with annotations of choledochal sections.

In contrast to the authors of [12], who propose a method for early detection of cholangiocarcinoma using hyperspectral images of microscopic tissues, using the ResNet-50 model, which achieves an accuracy of 82.4%. We additionally consider reducing the multidimensionality of the database using two methods: PCA and LDA. In addition, we parallelised these methods using modern parallel computing technologies Dask-ML and MPI, which allowed us to significantly improve data processing efficiency and diagnostic accuracy.

In [13], the authors use hyperspectral imaging (HSI), which offers a promising way to improve liver cancer diagnosis due to its ability to capture detailed continuous spectral and spatial information that is beyond the visible range of the human eye. Classification of cholangiocarcinoma using HSI is challenging due to its high dimensionality. To solve this problem, this article presents a network called MedisawHSI. As a result, they managed to achieve an accuracy of 93.35%. As we can see, the authors managed to achieve better accuracy compared to [12]. In our opinion, the accuracy of the results has improved because they used the division of the hyperspectral image into smaller overlapping regions, which are then classified individually based on their spectral characteristics.

In our study, we propose a method to enhance the accuracy and efficiency of cancer prognostication by reducing data dimensionality through PCA and LDA, parallelized using Dask-ML technology. This approach not only improves diagnostic accuracy but also significantly reduces preprocessing time. Similar to our efforts to address the challenges of time-consuming processes in medical data analysis, recent advancements in the Internet of Medical Things (IoMT) have also focused on improving the security and efficiency of data handling. For instance, a Timestamp-based

Secret Key Generation (T-SKG) scheme has been developed for resource-constrained IoMT devices to ensure secure data transmission without direct key sharing, thereby addressing vulnerabilities in traditional key sharing mechanisms [14]. This parallel development in secure data processing complements our efforts to enhance the reliability and speed of medical diagnostics.

The purpose of our article is to compare the efficiency of reducing the multidimensionality of the database using PCA and LDA methods applied at the preprocessing stage and parallelised using Dask-ML and MPI technologies to reduce processing time and improve the accuracy of data analysis in medicine, in particular, in the diagnosis of cholangiocarcinoma.

The main contributions of the paper are as follows:

- An improved approach to reducing data dimensionality using PCA and LDA methods is proposed, contributing to the accuracy of disease prediction.
- For the first time, Dask-ML technologies are used to parallelize PCA and LDA methods, significantly reducing data processing time.
- A comparative analysis of the performance and efficiency of PCA and LDA methods in reducing data dimensionality to enhance forecasting results is conducted.

2. Methods and materials

2.1. Overview of the algorithm of the proposed approach

The proposed approach consists of two parts and is schematically presented in Figure 1:

1. First of all, we applied data preprocessing to reduce the dimensionality of our multidimensional database. This will help us to reduce the dimensionality of the database while preserving the meaningful characteristics. We consider two methods of dimensionality reduction: PCA and LDA. To determine the effectiveness of each method, we propose to parallelise them and analyse the results. To parallelise the methods, we use the Dask-ML library [15]. Dask-ML is a toolkit that provides parallelised implementations of machine learning algorithms. Specifically, for PCA, we use Dask-ML PCA, which works with Dask Array to represent data and automatically distributes computations across multiple processors or computers.
2. Next, we will work with already processed data, namely, a smaller database after applying a dimensionality reduction method such as PCA or LDA. The RestNet-50 method is used for classification, to comparatively evaluate the effectiveness of the impact of reducing the dimensionality of the database as a way to solve an incorrectly posed problem in determining the diagnosis of cancer.

2.2. Overview of sequential PCA and LDA methods and their comparative characteristics

PCA is a statistical procedure that uses an orthogonal transformation. PCA transforms a group of correlated variables into a group of uncorrelated variables. Instead of discarding weak predictors, PCA generates new predictors that are uncorrelated. But in general, PCA works better if the data set contains independent but uncorrelated predictors, and another problem is the choice of the number of principal components [16]. The main goal of LDA is to project a dataset with a large number of features into a smaller space with good class resolution. This will reduce computational costs [17].

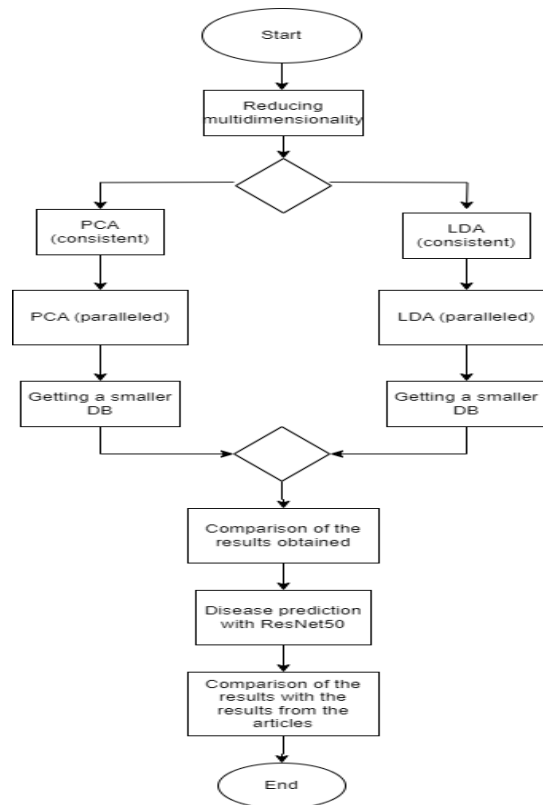


Figure 1: Diagram of the algorithm of the proposed approach

Dimensionality reduction techniques often require intensive computation and do not easily scale to large datasets. Recent advances in high-performance measurements using physical objects such as sensors or the results of complex numerical simulations generate data of extremely high dimensionality. It is becoming increasingly difficult to process such data consistently. In our research, we found that these methods were parallelised on distributed memory machines with MPI. The results show that their structure provides very good scalability for large problem sizes across the entire range of tested processor configurations [18], [19].

First of all, when applying the PCA method, we have to convert the data into feature vectors (we represent one image as one feature vector containing the pixels of the image). Thus, we have to standardise the data so that all features have the same weight.

An important step in understanding the relationships between attributes is the covariance matrix we build for standardised data. A covariance matrix is a square matrix that contains the covariances between all pairs of variables in your data set. Covariance measures how much two variables change together. Figure 2 shows a part of the covariance matrix with a size of 100×100 . In this case, we obtained:

1. The diagonal elements are equal to 1, indicating that each variable is perfectly correlated with itself. For standardised data, these values are always 1.
2. Off-diagonal elements: we can see different colours reflecting the level of covariance between different variables. Lighter colours (closer to yellow) indicate high positive covariance (variables that change in the same direction), while darker colours (closer to black) indicate low or negative covariance (variables that change in opposite directions).
3. The scale on the right shows how the colours of our matrix are interpreted: values closer to 1 indicate high positive covariance, and values closer to -0.2 indicate negative covariance. We can see that most of the matrix elements have positive covariance, which indicates that the variables in our dataset are likely to be correlated with each other.

Now we can calculate the eigenvectors and eigenvalues. We sort the eigenvectors in descending order of their eigenvalues in order to select the principal components for the algorithm. In our case, this number was 2 because we were reducing the database to two dimensions.

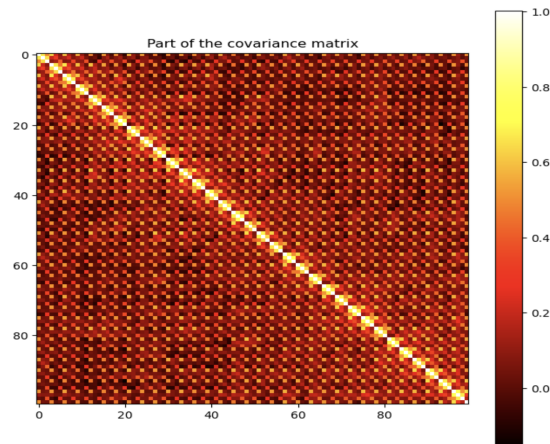


Figure 2: Part of the covariance matrix for standardised data

In Figure 3, each point represents one image from the database. Each axis represents the following:

1. "Height": the height of the image in pixels.
2. "Width": The width of the image in pixels.
3. "RGB value": This is the colour value of the image, which is represented in RGB channels.

Each dot in Figure 3 corresponds to one image. The colours of the dots differ depending on the folder to which they belong: 'L', 'N' or 'P'.

This figure allows you to visually understand the distribution of images in three-dimensional space in terms of their height, width, and color representation in RGB channels.

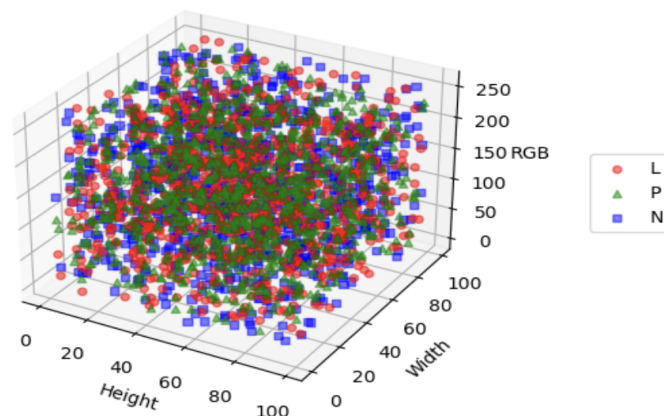


Figure 3: 3D model

As we mentioned before, the multidimensional choledoch database consists of images of three species. That is why we didn't reduce the multidimensionality of the entire database at once, but separately for L, P, N. As a result, we saved the reduced images in .h5 format. Figure 4 shows the original image.

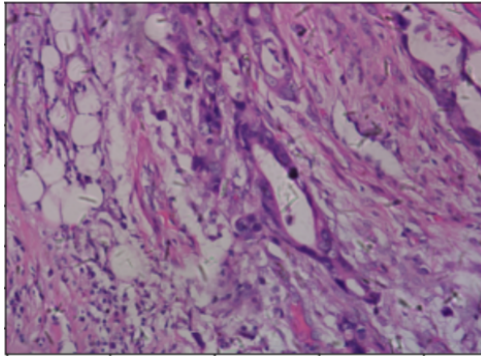


Figure 4: Original image

Figure 5 shows a representation of the relationship between the height and width of the images on a two-dimensional plane, and Figure 6 shows a reduced 2D image:

- Height: The height value of an image is in pixels. This is represented on the OX axis of the graph.
- Width: The width value of an image is in pixels. This is represented on the OY axis of the graph.

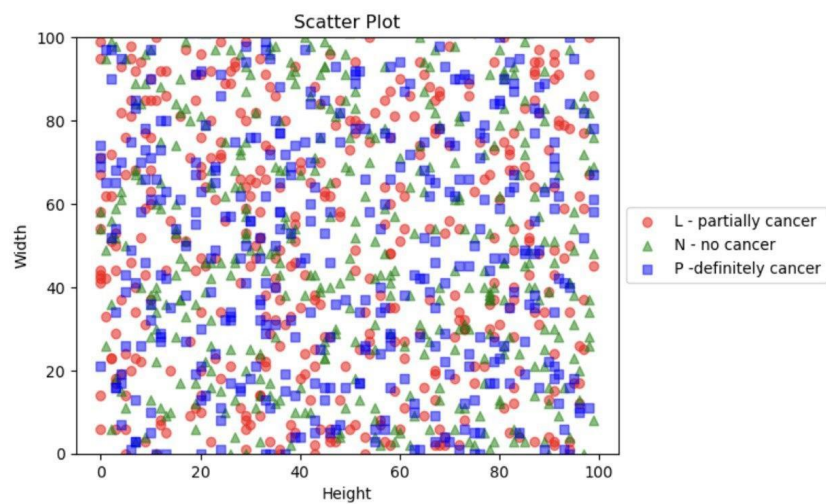


Figure 5: 2D model

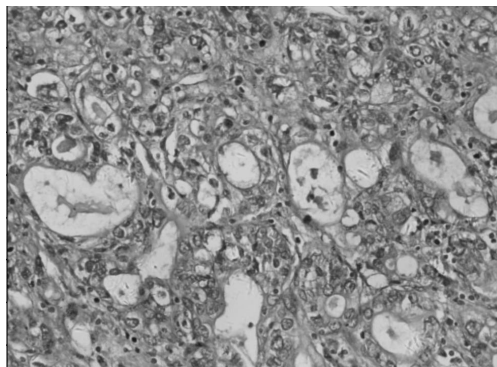


Figure 6: Reduced two-dimensional image

An overview of the key characteristics and differences between PCA and LDA is provided in Table 1.

Table 1
Comparative characteristics of PCA and LDA methods for data dimensionality reduction

| Criterion | PCA | LDA |
|--------------------------|--|---|
| Purpose | Dimensionality reduction by maximizing the total variance of the data | Reducing dimensionality by maximizing the difference between classes |
| Methodology | Eigenvectors and eigenvalues of the covariance matrix | Eigenvectors and eigenvalues of the scatter matrix between and within classes |
| Orientation | Independent of class | Class oriented |
| Application | Visualization, noise reduction, data preprocessing | Classification, improving the separation between classes |
| Data types | Any data | Class labeled data |
| Advantages | <ul style="list-style-type: none"> - Keeps the maximum amount of variation; - Useful for data visualization; - Independent of class | <ul style="list-style-type: none"> - Maximizes the separation between classes; - Effective for classification tasks; - Can improve classification results; |
| Disadvantages | <ul style="list-style-type: none"> - The loss of interpretation; - Not always suitable for classification tasks | <ul style="list-style-type: none"> - Assumption of data normality; - Loss of efficiency with a large number of classes or unequal covariance matrices; |
| Dimensionality reduction | To the number of principal components that retain most of the variance | To $(k-1)$, where k is the number of classes |
| Data requirements | Does not require class labels | Requires class labels and assumes a normal distribution of data with equal covariance matrices for each class |

2.3 Formal description of the parallel PCA and LDA algorithm using Dask-ML

The next stage of our research involves parallelizing PCA and LDA using the Dask-ML library [20], which allows us to scale computations to multiple processors or computers in a cluster. This is especially useful when working with large databases.

Stages of the parallel PCA algorithm using Dask-ML:

1. Calculation of the covariance matrix

- The input data is centered by subtracting the average value of each feature, which is represented by the formula (1)

$$X_{centered} = X - mean(X); \quad (1)$$

- The covariance matrix is calculated from the centered data (see (2))

$$\Sigma = \frac{1}{n-1} X_{centered}^T X_{centered}. \quad (2)$$

2. Calculation of Householder coefficients

- Based on the resulting covariance matrix, we calculate the Householder coefficient required to update the matrices Q and R , where Q is the orthogonal matrix from the QR decomposition and R is the upper triangular matrix from the QR decomposition;
- We calculate the norm of the vector v for its normalization, which is represented by the formula (3)

$$\|v\| = \sqrt{v_1^2 + v_2^2 + \dots + v_n^2}; \quad (3)$$

- Create a normalized vector u using the formula (4)

$$u = \frac{v}{\|v\|}; \quad (4)$$

- Calculate the coefficient β used to create the Householder reflection

$$\beta = \frac{2}{u^T u}. \quad (5)$$

3. Update matrices Q and R

- Divide the calculations of Q and R into parallel parts;
- Each column i of the matrix A is processed separately:
 - Calculate a part of the matrix Q (Qi);
 - Calculate a part of the matrix R (Ri);
- Combine the parts along the axis 1 (horizontally) to obtain the full matrices Q and R .

4. Calculating eigenvalues and eigenvectors

- Eigenvalues are calculated from the diagonal elements of the matrix R : $\lambda = \text{diag}(R)^2$;
- Eigenvectors are calculated by solving a system of linear equations using the Gaussian method for each eigenvalue: $Ae_i = \lambda_i e_i$.
- Parallelize this process to speed up the calculations:
 - Divide the array of eigenvalues into N subparts, where N is the number of threads.
 - At the same time, we calculate eigenvectors.

5. Conversion of eigenvectors to the basis A

- Eigenvectors obtained from the matrix R , are converted to the base A by multiplying by the matrix Q , as shown in (6)

$$e_A = Qe_R. \quad (6)$$

Stages of the parallel LDA algorithm using Dask-ML:

1. Use Dask-ML to calculate averages and center data in parallel;
2. Use Dask-ML to compute the mean vectors of each class in parallel;
3. Calculate scattering matrices for each class in parallel;
4. Using Dask-ML to calculate the interclass scattering matrix in parallel;
5. In parallel, we sum the scattering matrices for each class;
6. Using Dask-ML to calculate eigenvectors and eigenvalues in parallel.

2.4 Using the ResNet-50 architecture

During the analysis of the literature, several methods of disease prediction for the selected dataset proved to be effective, namely: ResNet-50, InceptionResNetV2, Random Forest, etc. We decided to focus on the ResNet-50 method for the following reasons:

- The authors of the article [12] used ResNet-50 in their research and achieved good results. This confirms the effectiveness of this method for our purposes.
- According to our analysis, the InceptionResNetV2 network was used by the authors of [9], but it was designed to work with multidimensional databases. Since our goal is to reduce the dimensionality of the database using parallel PCA and LDA algorithms, the InceptionResNetV2 method does not meet our needs.
- ResNet-50 is known to be an effective method for image classification and has successful results with different types of data. This makes it suitable for our task of predicting diseases from medical images.

Thus, when choosing ResNet-50, we took into account not only the availability of this method in the study, but also its suitability for our specific goals and limitations.

The authors of [12] emphasized the importance of data preparation, which included normalization and cropping of the original images, to achieve good results. We decided to follow these steps by reducing the size of all images, which confirms the adaptability of the method to our case.

To achieve the best performance, we split the dataset into non-overlapping training and test datasets, which are divided into training (6800 images) and test (210 images).

To determine the accuracy of the ResNet-50 neural network, we calculated how many tests the neural network gave correct answers and how many did not. We considered 4 cases to calculate the results:

- True Positive (TP) is a case where a person had cancer and the neural network gave the result that the person had cancer.
- True negative (TN) is a case where a person did not have cancer and the neural network determined correctly that the person really did not have cancer.
- False Negative (FN) is a case where a person had cancer, but the neural network said he did not.
- False Positive (FP) is a case where a person did not have cancer, but the neural network showed that they did.

To evaluate the forecasting efficiency, we chose the following metrics: Recall (the ratio of correctly identified positive cases to all actually positive cases), Precision (the ratio of correctly identified positive cases to all cases that the model identified as positive), and Accuracy (the ratio of correct predictions to the total number of observations). The formal representation of the latter is given by formulas (7)-(9).

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

3. Results of numerical experiments

Before presenting the results of the study, we propose to consider the characteristics of the computers on which the calculations were performed:

Computer models: Apple M1 Pro, Asus VivoBook

Operating systems: Windows 10, MacOS Sonoma 14.5

The amount of disk space available: 200 Гб

RAM: 16 Гб

Number of cores: 8

Network connection speed: 100 Мбit/c

Software tools we used to conduct our research: Clion, Kaggle editor, Google Colab.

In our work, we chose the choledoch dataset [21], which consists of three files:

- L – samples with parts of cancerous areas;
- N – complete cancerous areas;
- P – no cancerous areas;

Each file type includes three directories:

- annotation – contains the coordinates of the points where there are cancerous areas;

- hyper – general description of the image, including the number of columns, rows, channels, etc;
- rgb – contains all multidimensional images.

This dataset is multidimensional: DB dimensionality: (1728, 2304, 3), the first number indicates the height of the images, the second - the width, and the third - the number of dimensions.

Accordingly, separately for each type of image, we calculated the execution time of the sequential algorithms when reducing the multidimensional database using PCA and LDA methods and the total execution time. The results are presented in Table 2. The execution time of the program that implements the proposed parallel PCA and LDA methods depending on the number of cores is shown in Table 3.

Table 2
Time to execute sequential PCA and LDA methods, min

| Image type | L | N | P | Total time |
|----------------|------|------|------|------------|
| Sequential PCA | 7.01 | 6.43 | 6.57 | 20 |
| Sequential LDA | 7.10 | 7.23 | 7.02 | 21.53 |

Table 3
Execution time of parallel PCA and LDA methods depending on the number of cores, min

| Number of cores | 1 | 2 | 4 | 8 |
|-----------------|-------|-------|-------|-------|
| Parallel PCA | 5.012 | 3.343 | 2.112 | 2.005 |
| Parallel LDA | 5.451 | 3.020 | 2.343 | 2.151 |

As we can see from Tables 2 and 3, the proposed division into threads and subtasks allowed us to reduce the execution time based on a parallel algorithm, which is important at the preprocessing stage in order to reduce the dimensionality of data without significant time costs.

Next, it is important to analyze the overall diagnostic results. Table 4 shows the results of calculations of how many tests the neural network gave correct answers and how many did not.

Table 4
ResNet-50 test results

| Result | Positive | Negative |
|----------|----------|----------|
| Positive | 107 – TP | 3 – FN |
| Negative | 28 – FP | 72 – TN |

Next, we calculated the prediction accuracy indicators based on the proposed preprocessing stage and the use of the ResNet-50 network (see Table 5).

Table 5
Indicators of forecasting accuracy

| Result | Recall | Precision | Accuracy |
|----------|--------|-----------|----------|
| Positive | 0.793 | 0.972 | 0.852 |
| Negative | 0.722 | 0.965 | - |

The accuracy of our proposed method reached 85.2%, which is about 3% better than in [12]. This leads to the conclusion that data preprocessing and dimensionality reduction can avoid incorrectly set tasks and improve the accuracy of the results. At the same time, we also managed to solve the

problem of significant time costs at the preprocessing stage by parallelizing the PCA method using parallel computing technology.

4. Conclusions

Summarizing the results of this study, it should be emphasized that LDA is slower than PCA. We believe that this is due to the following factors: first, the calculation of the covariance matrix, as LDA needs to calculate the covariance matrix for each class in the dataset, which can be a computationally expensive operation, especially for large datasets with many classes. PCA, on the other hand, only needs to compute one covariance matrix for the entire dataset. Second, solving the eigenvalue problem, as LDA needs to solve the eigenvalue problem for the generalized eigenvalue matrix, which can be a computationally intensive task, especially for large matrices. PCA, on the other hand, requires solving the eigenvalue problem for a standardized covariance matrix, which is usually easier. Third, the number of eigenvalues: LDA typically requires only k eigenvalues to be calculated, where k is the desired projection dimension, while PCA requires all eigenvalues of the covariance matrix to be calculated. Fourth, sensitivity to noise: LDA can be more sensitive to noise in the data than PCA because LDA uses class label information, which can be sensitive to noise, while PCA does not use this information and therefore may be less sensitive to noise in the data. In general, LDA can be slower than PCA due to the more complex computations it requires. When applying the parallel PCA algorithm, we obtained a maximum speedup of about 3.5 times and an efficiency of 0.81. The dimensionality was reduced from three to two, which significantly improved the performance of our diagnostic system.

Acknowledgements

The authors express their gratitude to the Armed Forces of Ukraine for providing the security necessary to perform this work. This work has been made possible only through the resilience and courage of the Ukrainian Army.

References

- [1] D. Chumachenko, On Intelligent Multiagent Approach to Viral Hepatitis B Epidemic Processes Simulation, in 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), Lviv: IEEE, Aug. 2018, pp. 415–419. doi: 10.1109/DSMP.2018.8478602.
- [2] L. Mochurad and R. Panto, A Parallel Algorithm for the Detection of Eye Disease, in Advances in Intelligent Systems, Computer Science and Digital Economics IV, vol. 158, Z. Hu, Y. Wang, and M. He, Eds., in Lecture Notes on Data Engineering and Communications Technologies, vol. 158. , Cham: Springer Nature Switzerland, 2023, pp. 111–125. doi: 10.1007/978-3-031-24475-9_10.
- [3] A. Raza *et al.*, A Hybrid Deep Learning-Based Approach for Brain Tumor Classification, Electronics, vol. 11, no. 7, p. 1146, Apr. 2022, doi: 10.3390/electronics11071146.
- [4] M. Gitlin, N. McGarvey, N. Shivaprakash, and Z. Cong, Time duration and health care resource use during cancer diagnoses in the United States: A large claims database analysis, J. Manag. Care Spec. Pharm., vol. 29, no. 6, pp. 659–670, Jun. 2023, doi: 10.18553/jmcp.2023.29.6.659.
- [5] A. O. A. Deheyab *et al.*, AN OVERVIEW OF CHALLENGES IN MEDICAL IMAGE PROCESSING, in Proceedings of the 6th International Conference on Future Networks & Distributed Systems, Tashkent TAS Uzbekistan: ACM, Dec. 2022, pp. 511–516. doi: 10.1145/3584202.3584278.
- [6] S. Wang *et al.*, Advances in Data Preprocessing for Biomedical Data Fusion: An Overview of the Methods, Challenges, and Prospects, Inf. Fusion, vol. 76, pp. 376–421, Dec. 2021, doi: 10.1016/j.inffus.2021.07.001.

- [7] V. V. Bozhenko and T. M. Tatarnikova, Application of Data Preprocessing in Medical Research, in 2023 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF), St. Petersburg, Russian Federation: IEEE, May 2023, pp. 1–4. doi: 10.1109/WECONF57201.2023.10148004.
- [8] L. Mochurad and Y. Mochurad, Parallel Algorithms for Interpolation with Bezier Curves and B-Splines for Medical Data Recovery, 6th International Conference on Informatics and Data-Driven Medicine, IDDM 2023, vol. 3609, pp. 189–197, 2023.
- [9] Janarthhanan Jeyagopal, InceptionRestNetV2 Transfer Learning Approach for Cholangiocarcinoma Diagnosis utilizing Multidimensional Choledochal Database, 2021, Unpublished. doi: 10.13140/RG.2.2.21993.10083.
- [10] S. Nanga *et al.*, Review of Dimension Reduction Methods, J. Data Anal. Inf. Process., vol. 09, no. 03, pp. 189–231, 2021, doi: 10.4236/jdaip.2021.93013.
- [11] Q. Zhang, Q. Li, G. Yu, L. Sun, M. Zhou, and J. Chu, A Multidimensional Choledoch Database and Benchmarks for Cholangiocarcinoma Diagnosis, IEEE Access, vol. 7, pp. 149414–149421, 2019, doi: 10.1109/ACCESS.2019.2947470.
- [12] Y. Deng, J. Yin, Y. Wang, J. Chen, L. Sun, and Q. Li, ResNet-50 based Method for Cholangiocarcinoma Identification from Microscopic Hyperspectral Pathology Images, J. Phys. Conf. Ser., vol. 1880, no. 1, p. 012019, Apr. 2021, doi: 10.1088/1742-6596/1880/1/012019.
- [13] H. Namburu, V. N. Munipalli, M. Vanga, M. Pasam, S. Sikhakolli, and S. Chinnadurai, Cholangiocarcinoma Classification using MedisawHSI: A Breakthrough in Medical Imaging, in 2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE), Vellore, India: IEEE, Feb. 2024, pp. 1–6. doi: 10.1109/ic-ETITE58242.2024.10493579.
- [14] S. Saif, P. Das, S. Biswas, S. Khan, M. A. Haq, and V. Kovtun, A secure data transmission framework for IoT enabled healthcare, Heliyon, vol. 10, no. 16, Aug. 2024, doi: 10.1016/j.heliyon.2024.e36269.
- [15] S. Raschka, J. Patterson, and C. Nolet, Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence, Information, vol. 11, no. 4, p. 193, Apr. 2020, doi: 10.3390/info11040193.
- [16] P. Misra and A. S. Yadav, Impact of Preprocessing Methods on Healthcare Predictions, SSRN Electron. J., 2019, doi: 10.2139/ssrn.3349586.
- [17] G. T. Reddy *et al.*, Analysis of Dimensionality Reduction Techniques on Big Data, IEEE Access, vol. 8, pp. 54776–54788, 2020, doi: 10.1109/ACCESS.2020.2980942.
- [18] S. K. Samudrala, J. Zola, S. Aluru, and B. Ganapathysubramanian, Parallel Framework for Dimensionality Reduction of Large-Scale Datasets, Sci. Program., vol. 2015, pp. 1–12, 2015, doi: 10.1155/2015/180214.
- [19] J. A. J. Alsayaydeh, A. Aziz, A. I. A. Rahman, and et. al., DEVELOPMENT OF PROGRAMMABLE HOME SECURITY USING GSM SYSTEM FOR EARLY PREVENTION, ARPN Journal of Engineering and Applied Sciences, vol. 16, no. 1, pp. 88–97.
- [20] S. Raschka, J. Patterson, and C. Nolet, Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence, Information, vol. 11, no. 4, p. 193, Apr. 2020, doi: 10.3390/info11040193.
- [21] Microscopic Hyperspectral Choledoch Dataset. Jul. 20, 2024. [Online]. Available: <https://www.kaggle.com/datasets/ethelzq/multidimensional-choledoch-database>.