

Научная статья  
УДК 004.853  
<https://doi.org/10.24143/2072-9502-2025-1-17-27>  
EDN ALXXGH

---

## Анализ изображений графиков зависимостей физических величин

---

*Дмитрий Михайлович Коробкин*<sup>✉</sup>, *Сергей Алексеевич Фоменков*

*Волгоградский государственный технический университет,  
Волгоград, Россия, dkorobkin80@mail.ru*<sup>✉</sup>

---

**Аннотация.** Используемые на этапе концептуального проектирования физические эффекты, описанные в «первичных» источниках информации, таких как патенты, часто содержат изображения графиков зависимостей, связывающих физические величины входа и выхода. Анализ данной информации и ее использование для расширения описания физического эффекта является актуальной задачей. Описана разработка метода анализа графических изображений для классификации графиков зависимостей входных и выходных физических величин. Для этого требуется сформировать размеченный массив графиков зависимостей, а также провести вычислительные эксперименты для выявления наиболее эффективных архитектур нейросетевых моделей. Разработаны алгоритмы сегментации изображений графиков зависимостей, позволяющие избавиться от зашумляющих (для задачи классификации) частей рисунка, таких как оси координат, их обозначения, координатные сетки и т. д., на решении данной задачи проверена эффективность библиотек OpenCV и scikit-image. Сформированный размеченный массив содержит более 26 тыс. изображений графиков зависимостей. Разработан и программно реализован алгоритм классификации изображений графиков зависимостей по 9 классам (вогнутое увеличение, вогнутое уменьшение, выпуклое увеличение, выпуклое уменьшение, линейное увеличение, линейное уменьшение, постоянство, скачкообразное увеличение, скачкообразное уменьшение). По итогам работы можно сделать вывод, что все 3 метода классификации изображений (LSTM, CNN и ViT) показывают на тестовой выборке почти одинаковые результаты Accuracy, Precision, Recall, F1-Score, AUC-ROC – 98 %. При этом на произвольных изображениях из патентного массива точность анализа падает: у метода LSTM и ViT примерно на 10 %, а у CNN примерно на 2 %.

**Ключевые слова:** физический эффект, график зависимости, патент, сегментация изображений, классификация изображений

**Благодарности:** исследование выполнено за счет гранта Российского научного фонда № 23-21-00464, <https://rscf.ru/project/23-21-00464/>.

**Для цитирования:** Коробкин Д. М., Фоменков С. А. Анализ изображений графиков зависимостей физических величин // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. 2025. № 1. С. 17–27. <https://doi.org/10.24143/2072-9502-2025-1-17-27>. EDN ALXXGH.

Original article

---

## Image analysis of physical quantities dependencies graphs

---

*Dmitriy M. Korobkin*<sup>✉</sup>, *Sergey A. Fomenkov*

*Volgograd State Technical University,  
Volgograd, Russia, dkorobkin80@mail.ru*<sup>✉</sup>

---

**Abstract.** Physical effects used at the conceptual design stage, described in “primary” sources of information, such as patents, often contain images of dependency graphs linking physical input and output quantities. Analysis of this information and its use to expand the description of a physical effect is a relevant task. The development of a method for analyzing graphic images for classifying dependency graphs of input and output physical quantities is described. This requires forming a labeled array of dependency graphs, as well as conducting computational experiments to identify the most effective architectures of neural network models. Algorithms for segmenting images of dependency graphs have been developed, allowing one to get rid of noisy (for the classification task) parts of the figure, such as coordinate axes, their designations, coordinate grids, etc.), the effectiveness of the OpenCV and scikit-image libraries has been tested on solving this problem. The formed labeled array contains more than 26 thousand images of dependency graphs. An algorithm for clustering images of dependency graphs by 9 classes (concave increase, concave decrease, convex increase, convex decrease, linear increase, linear decrease, constancy, jump increase, jump-like decrease) has

been developed and implemented in software. Based on the results of the work, it can be concluded that all 3 methods of image clustering (LSTM, CNN and ViT) show almost the same results on the test dataset: Accuracy, Precision, Recall, F1-Score, AUC-ROC – 98%. At the same time, on arbitrary images from the patent array, the accuracy of the analysis decreases: for the LSTM and ViT methods by about 10%, and for CNN by about 2%.

**Keywords:** physical effect, dependence graph, patent, image segmentation, image classification

**Acknowledgment:** the study was supported by the grant from the Russian Science Foundation No. 23-21-00464, <https://rscf.ru/en/project/23-21-00464/>.

**For citation:** Korobkin D. M., Fomenkov S. A. Image analysis of physical quantities dependencies graphs. *Vestnik of Astrakhan State Technical University. Series: Management, computer science and informatics.* 2025;1:17-27. (In Russ.). <https://doi.org/10.24143/2072-9502-2025-1-17-27>. EDN ALXXGH.

## Введение

При проектировании новых технических систем в качестве одного из самых перспективных подходов видится использование синтеза физического принципа действия. Физический принцип действия – структура, отражающая взаимосвязь физических эффектов, в своей совокупности приводящих к выполнению функции технической системы. Сложилась признанная научная школа поискового конструирования на основе структурированных физических знаний (А. И. Половинкин, В. А. Камаев и др.) и «вручную» сформирован фонд из 1 200 физических эффектов [1, 2], который требует автоматизированной актуализации на основе самых современных физических знаний.

Часто описание физического эффекта в «первичных» источниках информации содержит графики зависимостей, связывающие физические величины входа и выхода физического эффекта. Анализ данной информации и ее использование для расширения описания физического эффекта позволит повысить качество синтеза физического принципа действия.

Таким образом, актуальной является разработка метода анализа графических изображений для классификации графиков зависимостей входных и выходных физических величин. Для этого требуется сформировать размеченный массив графиков зависимостей, а также провести вычислительные эксперименты для выявления наиболее эффективных архитектур нейросетевых моделей на основе критериев точности, полноты, F-меры, AUC-ROC [3].

*Целью работы* является повышение эффективности анализа графических изображений для расширения описаний физических эффектов на основе распознанных типов зависимостей физических величин.

Задачи:

- изучить методы и библиотеки сегментации изображений;
- изучить методы, модели и библиотеки классификации изображений;
- разработать модуль сегментации графиков зависимостей;
- сформировать размеченный массив изображений, содержащих графики зависимостей (линейное увеличение, постоянство, скачкообразное уменьшение и т. д.);

– разработать модуль классификации изображений графиков зависимостей на основе нейросетевых подходов;

– оценить эффективность разработанных модулей сегментации и классификации изображений на основе метрик точности, полноты, F-меры, AUC-ROC.

## Алгоритм сегментации изображений графиков зависимостей

Сегментация изображений позволяет избавиться от зашумляющих (для задачи классификации) частей рисунка, таких как оси координат, их обозначения, координатные сетки и т. д.), чтобы на изображении остался только график зависимости.

Был разработан общий алгоритм сегментации (рис. 1), позволяющий выбрать один из следующих способов: водоразделом на основе библиотеки OpenCV [4] и при помощи библиотеки scikit-image [5].

На первых общих этапах происходит проверка и корректировка формы изображений, конвертация цветных изображений в градации серого цвета. Далее в зависимости от выбранной модели алгоритм сегментации осуществляется двумя разными путями:

1. Сегментация графиков зависимостей на основе библиотеки OpenCV. Алгоритм включает в себя несколько этапов: бинаризацию изображений с помощью адаптивной пороговой сегментации (метод Оцу (Otsu) [6]), морфологические операции для улучшения выделения объектов, определение и маркировку передней и неизвестной областей; выделение и отрисовку контуров.

2. Сегментация графиков зависимостей на основе библиотеки scikit-image. Чтобы сегментировать изображение, проводится конвертация изображения в RGB, находятся контуры, из которых исключаются мелкие и остается самый большой. Алгоритм включает несколько этапов: применение размытия и адаптивной пороговой сегментации, морфологические операции для улучшения выделения объектов, поиск контура и создание пустой маски для графика, вычисление площади каждого контура и удаление малых контуров, сохранение масок.

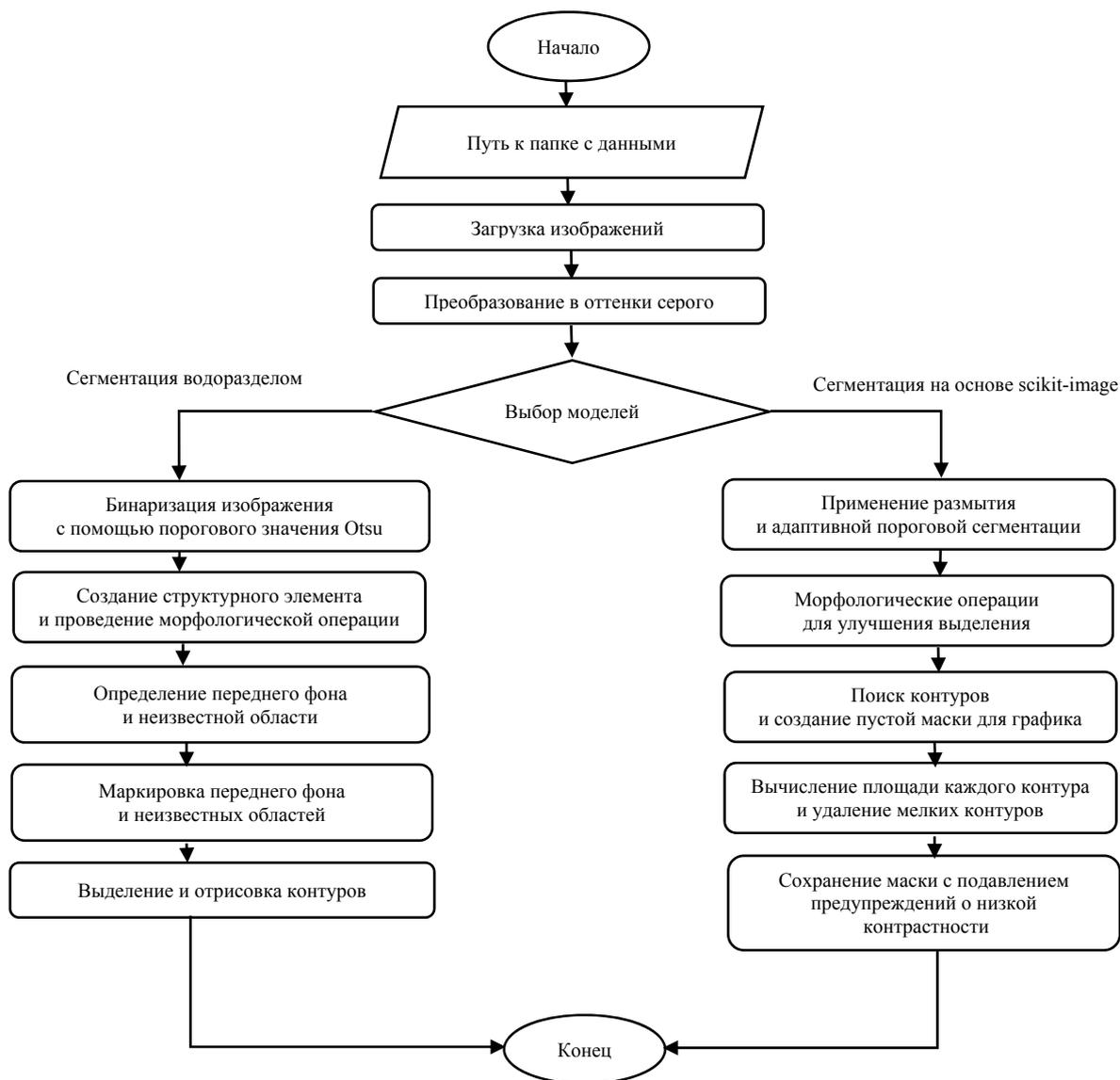


Рис. 1. Алгоритм сегментации изображений

Fig. 1. Image segmentation algorithm

Комбинирование этих инструментов и методов обеспечивает эффективный механизм сегментации графиков зависимостей, позволяющий обрабатывать изображения и выделять значимые объекты.

**Проверка эффективности сегментации графиков зависимостей на основе библиотеки scikit-image**

Для сегментации графиков зависимостей были использованы несколько ключевых инструментов. NumPy применяется для работы с массивами и числовыми операциями, что обеспечило эффективную обработку изображений и вычисление различных параметров, таких как площадь контуров. Библиоте-

ка scikit-image обеспечила основной функционал для обработки изображений. С помощью ее модулей `io`, `color`, `filters`, `measure` и `morphology` были реализованы такие задачи, как загрузка и сохранение изображений, конвертация RGB-изображений в градации серого, размытие изображений для уменьшения шума, адаптивная пороговая сегментация для выделения объектов, морфологические операции для улучшения выделения, поиск контуров и создание масок.

Результаты сегментации изображений на основе библиотеки `scikit-image` (`skimage`) представлены на рис. 2, 3.

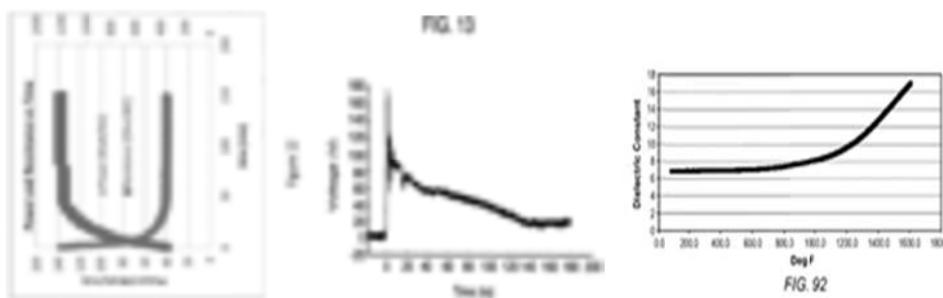


Рис. 2. Входные изображения для сегментации при помощи scikit-image

Fig. 2. Input images for scikit-image segmentation

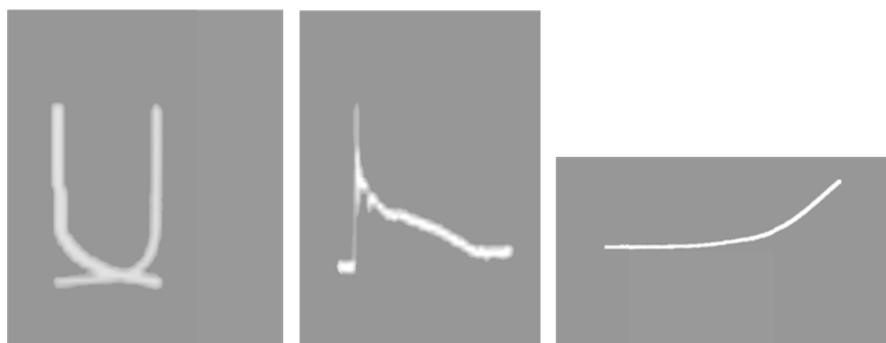


Рис. 3. Выходные изображения после сегментации при помощи scikit-image

Fig. 3. Output images for scikit-image segmentation

**Проверка эффективности сегментации графиков зависимостей на основе библиотеки компьютерного зрения OpenCV**

Были использованы следующие инструменты: NumPy применялся для работы с массивами и числовыми операциями; модуль Ipython.display использовался для отображения картинки; с помощью модуля cv2 библиотеки OpenCV были реализованы такие задачи, как конвертация RGB-изображений в градации серого, размытие изображений для уменьшения шума, адаптивная пороговая сегментация для выделения объектов, морфологические операции для улучшения выделения, поиск контуров и создание масок.

Было реализовано несколько методов сегментации:

1. Метод цветных пространств [7]. Цветные пространства представляют собой способ представления цветных каналов, присутствующих в изображении, который придает изображению определенный оттенок. Данный метод является одним из самых простых для сегментации изображения. Для улучшения сегментации изображения было применено гауссовское размытие – фильтр, использующий гауссовскую функцию для преобразования каждого пикселя изображения.

Результаты работы метода цветных пространств представлены на рис. 4, 5.

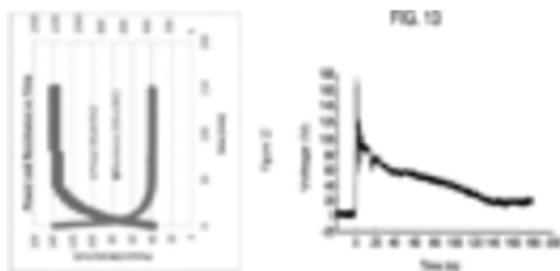


Рис. 4. Входные изображения для сегментации при помощи метода цветных пространств

Fig. 4. Input images for color-space conversion method



Рис. 5. Выходные изображения после сегментации при помощи метода цветных пространств

Fig. 5. Output images for color-space conversion methods

2. Метод watershed (водоразделения) [8]. Данный подход – классический метод сегментации изображений, основанный на концепции преобразования водораздела. В процессе сегментации будет учитываться сходство с соседними пикселями изображе-

ния в качестве важного ориентира для соединения пикселей с похожими пространственными позициями и значениями серого.

Результаты работы метода водораздела представлены на рис. 6, 7.

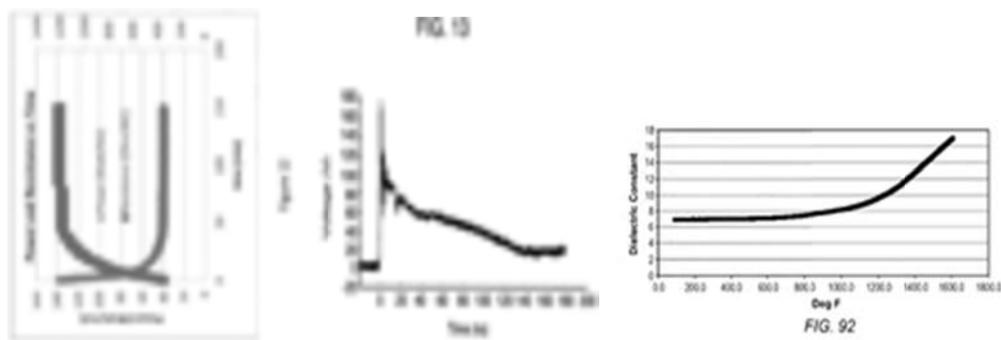


Рис. 6. Входные изображения для сегментации при помощи метода водоразделения

Fig. 6. Input images for watershed algorithm

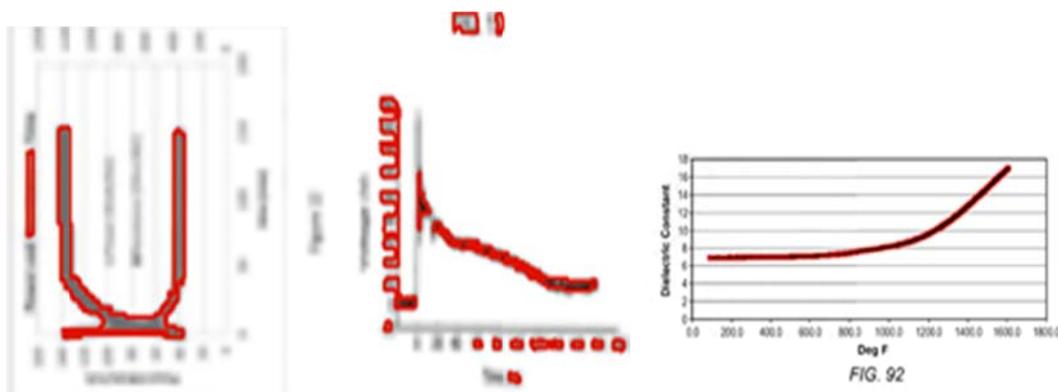


Рис. 7. Выходные изображения после сегментации при помощи метода водоразделения

Fig. 7. Output images for watershed algorithm

3. Метод выделения контура [9]. В данном методе находим контуры изображения, затем избав-

ляемся от шумов на изображении, находим контуры и выводим изображение (рис. 8, 9).

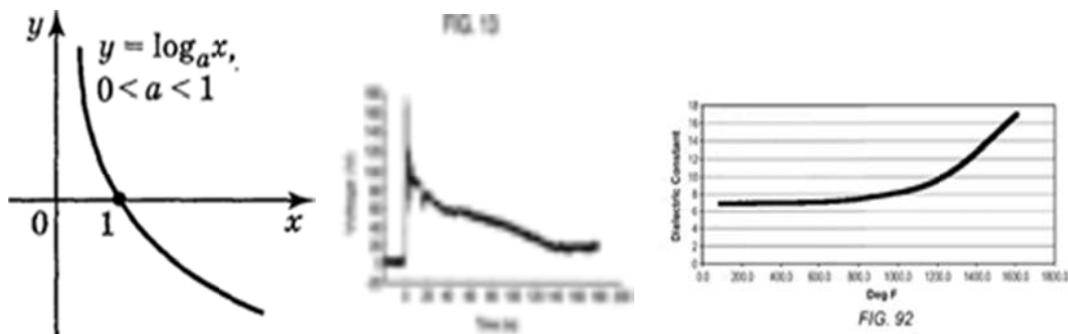


Рис. 8. Входные изображения для сегментации при помощи метода выделения контура

Fig. 8. Input images for countour finding

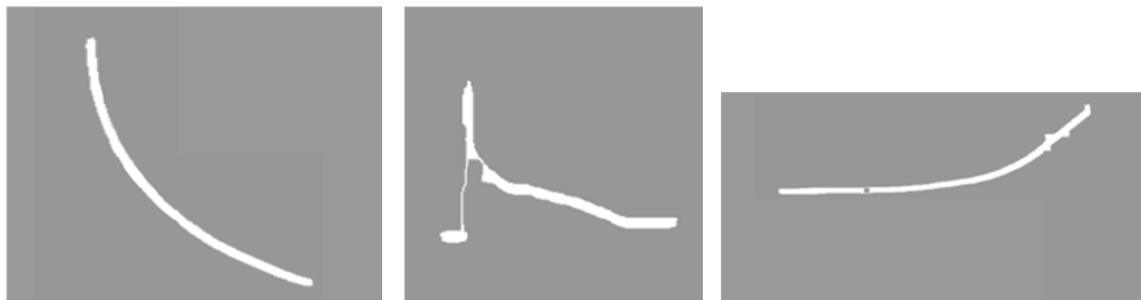


Рис. 9. Выходные изображения после сегментации при помощи метода выделения контура

Fig. 9. Output images for countour finding

Как видим, методы, реализованные на OpenCV, как правило, имеют более низкую точность сегментации, вследствие чего они менее надежны в сегментации графиков, хоть и достаточно точны. Из этого можно сделать вывод, что библиотека OpenCV и реализованные на ней классические методы сегментации уступают методу сегментации при помощи библиотеки scikit-image.

#### Алгоритм классификации изображений графиков зависимостей

В ходе работы была рассмотрена библиотека tensorflow [10] и методы LSTM [11], CNN [12], ViT [13], с помощью которых была реализована нейросеть для классификации графиков зависимостей по 9 классам (вогнутое увеличение, вогнутое уменьшение, выпуклое увеличение, выпуклое уменьшение, линейное увеличение, линейное уменьшение, постоянство, скачкообразное увеличение, скачкообразное уменьшение) (рис. 10).

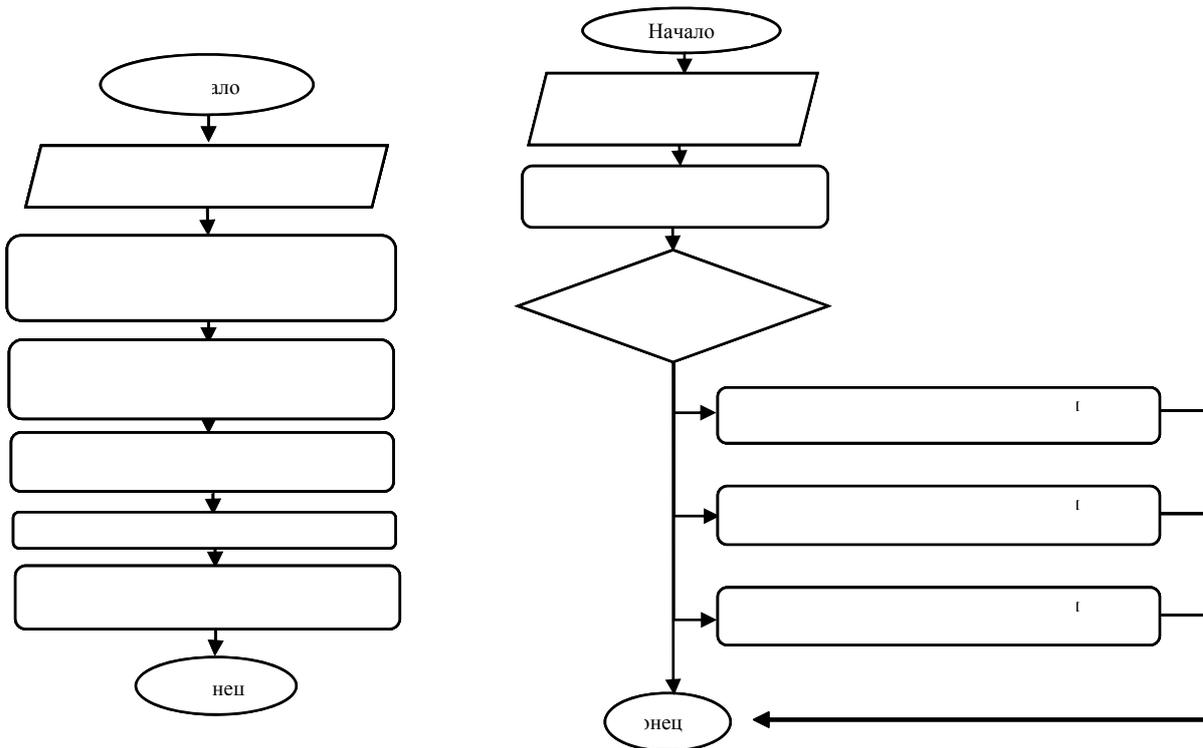


Рис. 10. Алгоритм классификации изображений

Fig. 10. Image classification algorithm

Также с помощью PyDesmos [15] и PyAutoGUI [16] осуществлялась генерация графиков и их графиче-

ский захват. PyDesmos – библиотека для работы с графическим калькулятором Desmos через его API.

PyAutoGUI – библиотека, которая используется для захвата изображений экрана. В коде применяется для создания скриншотов графиков, созданных с помощью PyDesmos. Скриншоты сохраняются в указанной директории с уникальными именами.

Сформированный при помощи библиотеки Matplotlib и PyDesmos датасет содержит более 26 тыс. изображений графиков зависимостей.

Модели обучаются на сформированных данных с использованием функции потерь SparseCategorical Crossentropy и оптимизатора Adam. После завершения обучения модель сохраняется в файл для последующего использования.

### Классификация графиков зависимостей с помощью метода LSTM

Для классификации использовались фреймворк TensorFlow и Keras – API для TensorFlow, упроща-

ющий создание и настройку нейронных сетей. Keras используется как для предварительной обработки данных, так и для создания архитектуры модели.

LSTM (Long Short-Term Memory) – это тип архитектуры рекуррентной нейронной сети (RNN), который обычно используется для задач моделирования последовательностей.

Для построения модели нейронной сети использовались сверточные слои (Conv2D), слой максимального объединения (MaxPooling2D), слои регуляризации (Dropout) и полносвязные слои (Flatten и Dense). Строятся графики точности и потерь на обучающей и валидационной выборках для анализа процесса обучения.

Обучена нейронная сеть, определяющая, какая зависимость представлена на изображении (рис. 11): доля верных ответов и ошибка обучения и валидации лежат в диапазоне от 0 до 1.

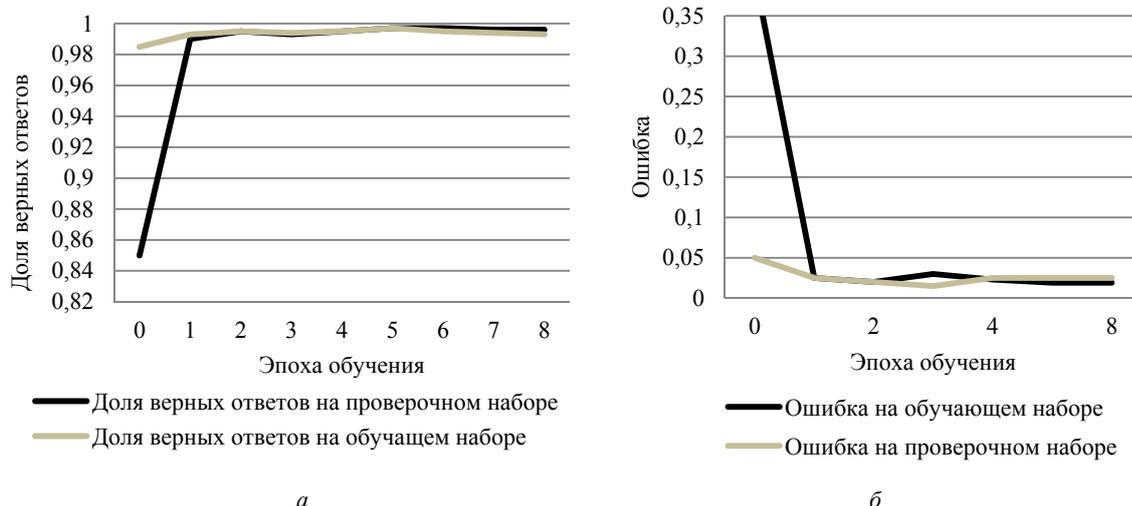


Рис. 11. Пример обучения при помощи модели LSTM:  
 а – точность обучения и валидации; б – потери обучения и валидации

Fig. 11. Example of training using an LSTM model: а – train and Test Accuracy; б – training and Validation errors

### Классификация графиков зависимостей с помощью метода CNN

Метод CNN (convolutional neural network, сверточная нейронная сеть) – тип нейронной сети, позволяющий обрабатывать данные с пространственной структурой, такие как изображения.

Для увеличения точности обучения, а также исправления проблемы переобучения модели была применена аугментация данных перед началом обучения. Аугментация данных (генерирование новых данных на основе имеющихся):

– `tf.keras.layers.experimental.preprocessing.Rescaling` –

слой для масштабирования значений пикселей изображений;

– `tf.keras.layers.experimental.preprocessing.RandomRotation`, `RandomZoom`, `RandomContrast` – слои для случайных преобразований изображений, таких как повороты, увеличение/уменьшение и изменение контраста, что помогает улучшить обобщающую способность модели.

Обучена нейронная сеть, определяющая, какая зависимость представлена на изображении (рис. 12): доля верных ответов и ошибка обучения и валидации лежат в диапазоне от 0 до 1.

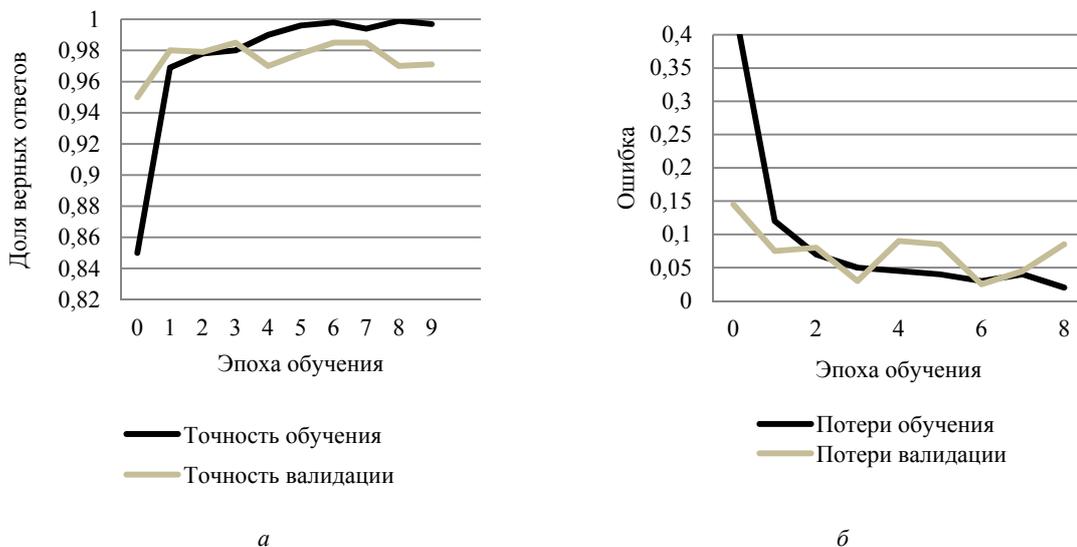


Рис. 12. Пример обучения при помощи модели CNN:  
 а – точность обучения и валидации; б – потери обучения и валидации

Fig. 12. Example of training using a CNN model: а – train and Test Accuracy; б – training and Validation errors

#### Классификация графиков зависимостей с помощью метода ViT

Метод ViT (Vision Transformer, трансформер для обработки изображений) – тип нейронной сети, использующий архитектуру трансформеров для анализа изображений. ViT способен обучаться сложным пространственным зависимостям в данных и позволяет значительно улучшить результаты в задачах компьютерного зрения.

Использовались:

– MobileNetV2 – предобученная модель для извлечения признаков из изображений, веса моде-

ли предварительно обучены на наборе данных ImageNet;

– аугментация данных: RandomFlip и RandomRotation – слои, применяемые к изображениям перед передачей их в предобученную модель, помогающие улучшить обобщающую способность модели за счет случайного поворота и отражения изображений.

Обучена нейронная сеть, определяющая, какая зависимость представлена на изображении (рис. 13): доля верных ответов и ошибка обучения и валидации лежат в диапазоне от 0 до 1.

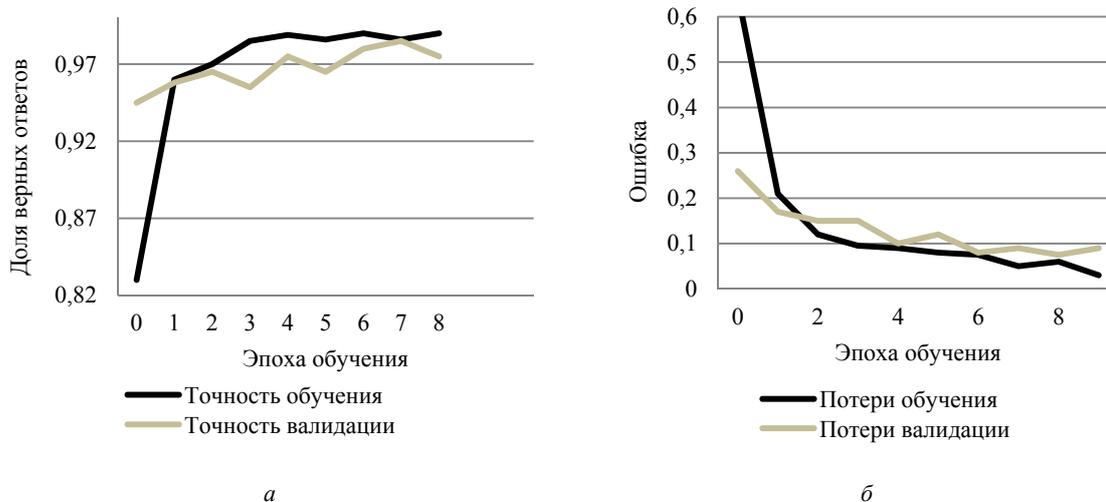


Рис. 13. Пример обучения при помощи модели ViT:  
 а – точность обучения и валидации; б – потери обучения и валидации

Fig. 13. Example of training using the ViT model: а – train and Test Accuracy; б – training and Validation errors

### Анализ эффективности моделей классификации изображений

Accuracy – это показатель, который описывает общую точность предсказания модели по всем классам. Это особенно полезно, когда каждый класс одинаково важен. Он рассчитывается как отношение количества правильных прогнозов к их общему количеству.

Precision характеризует долю правильно предсказанных релевантных классов среди всех образцов, которые модель спрогнозировала как релевантный класс.

Recall отражает долю правильно предсказанных релевантных классов среди всех реальных релевантных образцов.

F1 Score представляет собой гармоническое среднее между Precision и Recall, обеспечивая между ними баланс, что особенно полезно при неравномерном распределении классов.

В том случае, когда необходимо оценить качество модели при различных пороговых значениях, используется AUC-площадь (Area Under Curve) под ROC-кривой (Receiver Operating Characteristics), выраженной через отношение доли истинно релевантных прогнозов к доле нерелевантных.

Для вычисления метрик использовался модуль metrics. Для нахождения accuracy используется accuracy\_score(y\_true, y\_pred); precision – precision\_score(y\_true, y\_pred); recall – recall\_score(y\_true, y\_pred); f1-score – f1\_score(y\_true, y\_pred); ROC-AUC – roc\_auc\_score(y\_test, y\_pred), где y\_true – эталонные метки; y\_pred – предсказанные метки.

Метрики проверки эффективности методов показаны в таблице: в первой колонке метрики получены на тестовых данных, которых не было в обучающем датасете при обучении модели, во второй колонке – результаты на проверочном датасете с реальными изображениями графиков из текста патентов.

### Метрики эффективности

#### Performance metrics

Метод	На тестовом наборе	На произвольном наборе
LSTM	Validation Accuracy: 0.9844442004319022 Validation Precision: 0.9856259517640155 Validation Recall: 0.9844442004319022 Validation F1 Score: 0.9796998726648553 Validation ROC AUC: 0.9999767709888074	Validation Accuracy: 0.8982300884955752 Validation Precision: 0.9215481618976024 Validation Recall: 0.8982300884955752 Validation F1 Score: 0.8704177530852268 Validation ROC AUC: 0.9996753279542457
CNN	Validation Accuracy: 0.9823008849557522 Validation Precision: 0.9831544218712363 Validation Recall: 0.9823008849557522 Validation F1 Score: 0.9823999231765366 Validation ROC AUC: 0.9997857932559425	Validation Accuracy: 0.9623824451410659 Validation Precision: 0.9659810347621521 Validation Recall: 0.9623824451410659 Validation F1 Score: 0.9602090282137591 Validation ROC AUC: 1.0
ViT	Validation Accuracy: 0.9823008849557522 Validation Precision: 0.9831544218712363 Validation Recall: 0.9823008849557522 Validation F1 Score: 0.9823999231765366 Validation ROC AUC: 0.9997857932559425	Validation Accuracy: 0.876265466816648 Validation Precision: 0.8110381515505287 Validation Recall: 0.876265466816648 Validation F1 Score: 0.8348252704983662 Validation ROC AUC: 0.9971783748849872

После анализа метрик можно сделать вывод, что все 3 метода показывают почти одинаковые результаты при тестировании на сгенерированных изображениях – примерно 98 %, однако при тестировании на патентных изображениях точность анализа падает: у метода LSTM и ViT примерно на 10 %, а CNN примерно на 2 %, из чего можно сделать вывод, что для реальных изображений лучше всего использовать метод CNN.

### Заключение

Новизна разработанного метода анализа графических изображений для расширения описаний физических эффектов заключается в применении моделей глубокого обучения для распознавания типов зависимостей физических величин.

В результате были разработаны и программно

реализованы алгоритмы сегментации изображений графиков зависимостей, позволяющие избавиться от зашумляющих (для задачи классификации) частей рисунка, таких как оси координат, их обозначения, координатные сетки и т. д., на решении данной задачи проверена эффективность библиотек OpenCV и scikit-image. Сформирован размеченный массив графиков зависимостей, который содержит более 26 тыс. изображений графиков зависимостей. Разработан и программно реализован алгоритм классификации изображений графиков зависимостей по 9 классам (вогнутое увеличение, вогнутое уменьшение, выпуклое увеличение, выпуклое уменьшение, линейное увеличение, линейное уменьшение, постоянство, скачкообразное увеличение, скачкообразное уменьшение).

### Список источников

1. Коробкин Д. М. Архитектура автоматизированной системы актуализации физических знаний с использованием ClickHouse и HDFS // Информационные технологии. 2024. Т. 30. № 9. С. 467–473. DOI: 10.17587/it.30.467-473.
2. Козина С. А., Коробкин Д. М., Фоменков С. А., Колесников С. Г. Архитектура системы формирования единой базы знаний по физической тематике // Прикаспийск. журн.: управление и высокие технологии. 2023. № 1 (61). С. 27–37. DOI: 10.54398/20741707\_2023\_1\_27.
3. Powers D. V. W., Ailab. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation // J. Mach. Learn. Technol. 2011. V. 2 (1). P. 2229–3981. DOI: 10.9735/2229-3981.
4. Culjak I., Abram D., Pribanic T., Džapo H., Cifrek M. A brief introduction to OpenCV // 2012 Proceedings of the 35th International Convention MIPRO. 2012. P. 1725–1730.
5. Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., Duchesnay E., Louppe G. Scikit-learn: Machine Learning in Python // Journal of Machine Learning Research. 2011. V. 12. P. 2825–2830.
6. Xu Xiangyang, Xu Shengzhou, Jin Lianghai, Song Enmin. Characteristic analysis of Otsu threshold and its applications // Pattern Recognition Letters. 2011. V. 32. P. 956–961. DOI: 10.1016/j.patrec.2011.01.021.
7. Abdelsadek Dena, Al-berry Maryam, Ebied Hala, Hasaan Mosab. Impact of Using Different Color Spaces on the Image Segmentation // The 8th International Conference on Advanced Machine Learning and Technologies and Applications (AMTLA2022). 2022. P. 456–471. DOI: 10.1007/978-3-031-03918-8\_39.
8. Image Segmentation with Watershed Algorithm. URL: [https://docs.opencv.org/4.x/d3/db4/tutorial\\_py\\_water-shed.html](https://docs.opencv.org/4.x/d3/db4/tutorial_py_water-shed.html) (дата обращения: 12.09.2024).
9. Puri Raghav, Gupta Archit. Contour, Shape & Color Detection using OpenCV-Python // International Journal of Advances in Electronics and Computer Science. 2018. V. 5. Iss. 3. P. 26–27.
10. Ünsalan C., Höke B., Atmaca E. The TensorFlow Platform and Keras API // Embedded Machine Learning with Microcontrollers. Springer, Cham., 2025. P. 183–197. [https://doi.org/10.1007/978-3-031-69421-9\\_9](https://doi.org/10.1007/978-3-031-69421-9_9).
11. Tejasree Ganji, Loganathan Agilandeeswari. An extensive review of hyperspectral image classification and prediction: techniques and challenges // Multimedia Tools and Applications. 2024. V. 83. P. 80941–81038. DOI: 10.1007/s11042-024-18562-9.
12. Feng Zhenpeng, Ji Hongbing, Dakovic Milos, Cui Xiyang, Zhu Mingzhe, Stankovic Ljubisa. Cluster-CAM: Cluster-Weighted Visual Interpretation of CNNs' Decision in Image Classification. URL: <https://arxiv.org/abs/2302.01642> (дата обращения: 12.09.2024).
13. Shao Ran, Bi Xiao-Jun, Chen Zheng. Hybrid ViT-CNN Network for Fine-Grained Image Classification // IEEE Signal Processing Letters. 2024. P. 1–5. DOI: 10.1109/LSP.2024.3386112.
14. Hunter J. Matplotlib: A 2D Graphics Environment // Computing in Science & Engineering. 2007. V. 9. P. 90–95. DOI: 10.1109/MCSE.2007.55.
15. PyDesmos 0.1.3. URL: <https://pypi.org/project/PyDesmos/> (дата обращения: 12.09.2024).
16. Welcome to PyAutoGUI's documentation! URL: <https://pyautogui.readthedocs.io/en/latest/> (дата обращения: 12.09.2024).

### References

1. Korobkin D. M. Arhitektura avtomatizirovannoj sistemy aktualizacii fizicheskih znanij s ispol'zovaniem ClickHouse i HDFS [Architecture of an automated system for updating physical knowledge using ClickHouse and HDFS]. *Informacionnye tekhnologii*, 2024, vol. 30, no. 9, pp. 467-473. DOI: 10.17587/it.30.467-473.
2. Kozina S. A., Korobkin D. M., Fomenkov S. A., Kolesnikov S. G. Arhitektura sistemy formirovaniya edinoj bazy znanij po fizicheskoj tematike [Architecture of the system for forming a unified knowledge base on physical subjects]. *Pri-kaspijskij zhurnal: upravlenie i vysokie tekhnologii*, 2023, no. 1 (61), pp. 27-37. DOI: 10.54398/20741707\_2023\_1\_27.
3. Powers D. V. W., Ailab. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation. *J. Mach. Learn. Technol.*, 2011, vol. 2 (1), pp. 2229-3981. DOI: 10.9735/2229-3981.
4. Culjak I., Abram D., Pribanic T., Džapo H., Cifrek M. A brief introduction to OpenCV. *2012 Proceedings of the 35th International Convention MIPRO*, 2012, pp. 1725-1730.
5. Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., Duchesnay E., Louppe G. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 2011, vol. 12, pp. 2825-2830.
6. Xu Xiangyang, Xu Shengzhou, Jin Lianghai, Song Enmin. Characteristic analysis of Otsu threshold and its applications. *Pattern Recognition Letters*, 2011, vol. 32, pp. 956-961. DOI: 10.1016/j.patrec.2011.01.021.
7. Abdelsadek Dena, Al-berry Maryam, Ebied Hala, Hasaan Mosab. Impact of Using Different Color Spaces on the Image Segmentation. *The 8th International Conference on Advanced Machine Learning and Technologies and Applications (AMTLA2022)*, 2022, pp. 456-471. DOI: 10.1007/978-3-031-03918-8\_39.
8. *Image Segmentation with Watershed Algorithm*. Available at: [https://docs.opencv.org/4.x/d3/db4/tutorial\\_py\\_water-shed.html](https://docs.opencv.org/4.x/d3/db4/tutorial_py_water-shed.html) (accessed: 12.09.2024).
9. Puri Raghav, Gupta Archit. Contour, Shape & Color Detection using OpenCV-Python *International Journal of Advances in Electronics and Computer Science*, 2018, vol. 5, iss. 3, pp. 26-27.
10. Ünsalan C., Höke B., Atmaca E. *The TensorFlow Platform and Keras API. Embedded Machine Learning with Microcontrollers*. Springer, Cham., 2025. Pp. 183-197. [https://doi.org/10.1007/978-3-031-69421-9\\_9](https://doi.org/10.1007/978-3-031-69421-9_9).
11. Tejasree Ganji, Loganathan Agilandeeswari. An extensive review of hyperspectral image classification and prediction: techniques and challenges. *Multimedia Tools and Applications*, 2024, vol. 83, pp. 80941-81038. DOI: 10.1007/s11042-024-18562-9.

12. Feng Zhenpeng, Ji Hongbing, Dakovic Milos, Cui Xiyang, Zhu Mingzhe, Stankovic Ljubisa. *Cluster-CAM: Cluster-Weighted Visual Interpretation of CNNs' Decision in Image Classification*. Available at: <https://arxiv.org/abs/2302.01642> (accessed: 12.09.2024).

13. Shao Ran, Bi Xiao-Jun, Chen Zheng. Hybrid ViT-CNN Network for Fine-Grained Image Classification. *IEEE Signal Processing Letters*, 2024, pp. 1-5. DOI: 10.1109/LSP.2024.3386112.

14. Hunter J. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 2007, vol. 9, pp. 90-95. DOI: 10.1109/MCSE.2007.55.

15. *PyDesmos 0.1.3*. Available at: <https://pypi.org/project/PyDesmos/> (accessed: 12.09.2024).

16. *Welcome to PyAutoGUI's documentation!* Available at: <https://pyautogui.readthedocs.io/en/latest/> (accessed: 12.09.2024).

Статья поступила в редакцию 12.11.2024; одобрена после рецензирования 06.12.2024; принята к публикации 15.01.2025  
The article was submitted 12.11.2024; approved after reviewing 06.12.2024; accepted for publication 15.01.2025

### Информация об авторах / Information about the authors

**Дмитрий Михайлович Коробкин** – кандидат технических наук, доцент; доцент кафедры систем автоматизированного проектирования и поискового конструирования; Волгоградский государственный технический университет; [dkorobkin80@mail.ru](mailto:dkorobkin80@mail.ru)

**Dmitriy M. Korobkin** – Candidate of Technical Sciences, Assistant Professor; Assistant Professor of the Department of Computer-aided Design and Search Engineering; Volgograd State Technical University; [dkorobkin80@mail.ru](mailto:dkorobkin80@mail.ru)

**Сергей Алексеевич Фоменков** – доктор технических наук, профессор; профессор кафедры систем автоматизированного проектирования и поискового конструирования; Волгоградский государственный технический университет; [saf@vstu.ru](mailto:saf@vstu.ru)

**Sergey A. Fomenkov** – Doctor of Technical Sciences, Professor; Professor of the Department of Computer-aided Design and Search Engineering; Volgograd State Technical University; [saf@vstu.ru](mailto:saf@vstu.ru)

