

12. Leonenkov A.V. Nechetkoe modelirovanie v srede MATLAB i fuzzyTECH [Fuzzy modeling in MATLAB and FuzzyTECH]. Saint Petersburg: BKhV-Peterburg, 2003.
13. Ash T. Dynamic Node Creation in Back-Propagation Networks, *II Connection Science*, 1989, Vol. 1.
14. Kwan H., Cai L. A fuzzy neural network and its application to pattern recognition // IEEE Transactions on Fuzzy Systems. – 1994. – Vol. 2, No. 3. – P. 185-193. – DOI: 10.1109/91.298447.
15. Allen J.F. Maintaining knowledge about temporal intervals, *Communications of the ACM*, 1983, Vol. 26, No. 11, pp. 832-843.
16. Wu J. and Yan S. Reliability Evaluation for Mechanical Systems by Petri Nets, *Petri Nets in Science and Engineering*, 2018.
17. Balbiani P., Boudou J., Diéguez M., Fernández-Duque D. Intuitionistic Linear Temporal Logics, *ACM Transactions on Computational Logic*, 2019, 21, pp. 1-32.
18. Vajnilovich Yu., Zaharchenkov K. and Zakharova A. Integrated System Approach to Improving the Efficiency of IT Projects Management Based on Evolutionary Modeling, *Computing in Physics and Technology*, pp. 309-314.
19. Borisov V.V., Misnik A.E. Ontological Engineering of Interrelated Processes in Complex Cyber-Physical Systems, In: Kovalev, S., Sukhanov, A., Akperov, I., Ozdemir, S. (eds), *Proceedings of the Sixth International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'22). IITI 2022. Lecture Notes in Networks and Systems*, Vol. 566. Springer, Cham, 2023. Available at: https://doi.org/10.1007/978-3-031-19620-1_39.
20. Karabach A. Information integration systems based on semantic technologies, *Science, technology and education*, 2014, Vol. 2 (2).

Прокопенко Сергей Александрович – Белорусско-Российский университет; e-mail: puss95@yandex.by; г. Могилев, Беларусь; тел.: +375447535081; +79679872264; ассистент; ассистент НИУ «МЭИ».

Бобряков Александр Владимирович – НИУ «МЭИ»; e-mail: avbob@mail.ru; г. Москва, Россия; тел.: +74953627407; д.т.н.; зав. кафедрой УИТ.

Prokopenko Sergey Aleksandrovich – Belarusian-Russian University; e-mail: puss95@yandex.by; Mogilev, Belarus; phones: +375447535081; +79679872264; assistant; assistant of National Research University "Moscow Power Engineering Institute".

Bobryakov Alexander Vladimirovich – National Research University "Moscow Power Engineering Institute"; e-mail: avbob@mail.ru; Moscow, Russia; phone: +74953627407; dr. of eng. sc.; head of the Department of Information Technologies;

УДК 004.932

DOI 10.18522/2311-3103-2026-1-284-295

В.Е. Бондарева, Т.С. Черноморова, А.В. Кривцун, А. Абдулкарем

АЛГОРИТМ РАСПОЗНАВАНИЯ ЯЗЫКА ЖЕСТОВ С ИСПОЛЬЗОВАНИЕМ СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ

Рассматривается задача автоматического распознавания русского жестового языка (РЖЯ) на основе методов компьютерного зрения и глубокого обучения. Актуальность исследования обусловлена устойчивым ростом численности лиц с нарушениями слуха: по данным Всемирной организации здравоохранения, в настоящее время их количество составляет около 70 млн человек, и, согласно прогнозам, может увеличиться до 630 млн к 2035 году. Разработка эффективных алгоритмов распознавания жестов является важным направлением создания систем бесконтактного человеко-машинного взаимодействия, направленных на повышение доступности информационных технологий и улучшение качества жизни людей с ограниченными возможностями слуха. Целью работы является разработка и экспериментальная апробация алгоритма распознавания букв русского жестового языка в видеопотоке в реальном времени с использованием сверточной нейронной сети. В ходе исследования сформирован специализированный датасет, включающий 430 изображений жестов, соответствующих буквам алфавита РЖЯ, представленных с различных ракурсов и при изменяющихся условиях освещения. Для реализации модели использованы библиотеки TensorFlow и Keras, а интеграция с видеопотоком выполнена с применением средств OpenCV и системы отслеживания положения руки на основе маркеров. В результате проведенного обучения и тестирования предложенная модель продемонстрировала точность распознавания на уровне 99

% на тестовой выборке. Проведён сравнительный анализ с классическими методами машинного обучения, показавший преимущество сверточной нейронной сети по точности классификации и устойчивости к внешним помехам. Полученные результаты подтверждают эффективность предложенного подхода и его применимость для создания систем реального времени, предназначенных для коммуникационных, образовательных и реабилитационных приложений, а также для разработки интерфейсов человеко-машинного взаимодействия.

Язык жестов; сверточная нейронная сеть; машинное обучение.

V.E. Bondareva, T.S. Chernomorova, A.V. Krivtsun, A. Abdulkarem

ALGORITHM FOR SIGN LANGUAGE RECOGNITION USING CONVOLUTIONAL NEURAL NETWORK

This paper addresses the problem of automatic recognition of Russian Sign Language (RSL) using computer vision and deep learning methods. The relevance of the study is driven by a steady increase in the number of people with hearing impairments: according to the World Health Organization, there are currently about 70 million deaf and hard-of-hearing individuals worldwide, and this number is projected to reach 630 million by 2035. The development of effective gesture recognition algorithms is an important direction for creating contactless human-machine interaction systems aimed at improving accessibility of information technologies and enhancing the quality of life for people with hearing disabilities. The aim of the study is to develop and experimentally validate an algorithm for real-time recognition of Russian Sign Language alphabet gestures in a video stream using a convolutional neural network. A specialized dataset was created, consisting of 430 images of hand gestures corresponding to the letters of the RSL alphabet, captured from different angles and under varying lighting conditions. The model was implemented using TensorFlow and Keras libraries, while integration with the video stream was performed using OpenCV and a marker-based hand tracking system. As a result of training and testing, the proposed model achieved a recognition accuracy of 99% on the test dataset. A comparative analysis with classical machine learning methods demonstrated the superiority of the convolutional neural network in terms of classification accuracy and robustness to external noise. The obtained results confirm the effectiveness of the proposed approach and its applicability for real-time systems intended for communication, educational, and rehabilitation applications, as well as for the development of advanced human-machine interaction interfaces.

Sign language; convolutional neural network; machine learning.

Введение. К концу 2023 года по всему миру насчитывалась около 70 миллионов глухонемых людей. К сожалению, эта цифра продолжит расти и в будущем, так как потеря слуха может произойти не только из-за болезни, но и из-за внешних факторов, как громкие и постоянные звуки на протяжении длительного времени. По предварительным подсчётам количество глухонемых может возрасти до 630 миллионов человек к 2035 и до 900 миллионов человек к 2050 году [1]. Люди с нарушениями слуха общаются на языке жестов. Язык жестов состоит из множества символов, которые для каждой страны разные [2].

Компьютерное зрение помогает распознавать жесты, но для этого необходимо разработать и обучить специальные алгоритмы и нейросети. Для компьютерного зрения руки являются сложным объектом. На руках нет "активных" точек, подобно глазам и рту на лице, что затрудняет естественное восприятие языка жестов [3]. Для реализации подобных систем существует два основных подхода:

◆ **Маркерный подход.** В этом подходе на руки человека наносятся маркеры, которые могут быть оптическими, магнитными или акустическими. Камера, установленная на устройстве, считывает эти маркеры и определяет жесты [4].

◆ **Безмаркерный подход.** Этот подход использует алгоритмы компьютерного зрения для анализа движений рук и пальцев без использования дополнительных маркеров. Нейросеть анализирует жесты, используя алгоритмы компьютерного зрения и методы обработки изображений. Программа получает изображение с камеры, обрабатывает его и определяет различные параметры, такие как положение, ориентация и форма жеста. Затем нейросеть классифицирует жест на основе полученных данных и сравнивает его с известными жестами, хранящимися в её базе знаний. В результате модель может определить, что означает данный жест [5].

Сравнительные исследования показывают преимущества сверточных нейронных сетей (CNN) над классическими искусственными нейронными сетями (ИНС) за счёт эффективного выявления пространственных и временных признаков. Искусственная нейронная сеть (ИНС) – классическая сеть, в которой каждый нейрон связан с каждым на следующем слое; эффективно работает с векторизованными признаками, но не учитывает пространственную структуру изображений. Сверточная нейронная сеть (CNN) – специализированная архитектура для работы с изображениями; использует свертки для выявления локальных признаков, что обеспечивает высокую точность распознавания при анализе пространственно-временных паттернов жестов [6]. CNN обладают уникальной архитектурой, которая позволяет им эффективно выявлять пространственные признаки в изображениях. Это достигается за счет использования сверток, которые позволяют обнаруживать локальные паттерны и текстуры, что особенно важно для анализа движений рук и пальцев. Благодаря этому, CNN могут точно распознавать формы и положение объектов, что критично для задач, связанных с жестами и манипуляциями. [7]. Для анализа движений рук важна не только пространственная информация, но и временная динамика. Хотя традиционные CNN не предназначены для обработки последовательностей данных, их можно комбинировать с другими архитектурами, такими как долгосрочная краткосрочная память (LSTM) или временные сверточные сети (TCN). Это позволяет эффективно захватывать изменения во времени и улучшать точность распознавания движений [8]. CNN также показывают высокую устойчивость к шуму в данных. Это особенно важно в реальных условиях, где изображения могут быть искажены или содержать лишнюю информацию. За счет своей способности к обобщению и извлечению ключевых признаков из сложных входных данных, CNN обеспечивают надежность в распознавании жестов даже при наличии помех. Сверточные нейронные сети легко масштабируются для работы с различными размерами входных данных и могут быть адаптированы под конкретные задачи. Это делает их универсальным инструментом для разработки систем анализа движений рук и пальцев, которые могут быть использованы в различных приложениях – от виртуальной реальности до реабилитации. Кроме того, сверточные сети могут быть более простыми и эффективными в сравнении с полносвязными сетями, что делает их более подходящими для реализации на устройствах с ограниченными ресурсами.

Современные исследования показывают актуальность выбранной темы исследования. Исследование Байхана и соавторов (2024) предложило модифицированную модель CNNsa-LSTM, которая объединяет сверточную нейронную сеть (CNN) для извлечения пространственных признаков, механизм самовнимания (Self-Attention, SA) для выделения ключевой информации и рекуррентную LSTM для моделирования временной динамики жестов. Такая архитектура позволила достичь высокой точности – около 96,4% – при классификации жестов языка глухонемых на видео. Использование BLSTM (двухнаправленной LSTM) улучшает устойчивость и качество распознавания продолжительных сигналов, важных для перевода сложных жестов [9]. Кумар с коллегами (2022) реализовали систему Markerless hand gesture recognition (Безмаркерное распознавание с глубоким обучением), которая использует методы глубокого обучения и сверточные сети для сегментации и анализа движений рук в реальном времени. Их алгоритм показал высокую точность классификации даже в сложных условиях освещения и фона, что подтверждает потенциал безмаркерных методов в практическом применении для распознавания языка жестов [10]. Работа Амёра и соавторов (2025) представила сложную архитектуру пространственно-временной модели с генеративной сетью, сочетающую deep LSTM autoencoder и рекуррентные сети, что позволило не только классифицировать жесты, но и восстанавливать последовательности для генерации дополнительных обучающих данных. Это увеличивает обобщающую способность модели и устойчивость к шуму, что критично для реального применения в распознавании жестов [11]. Аль-Мудави и коллеги (2024) разработали комплексную систему для распознавания жестов в реальных условиях, применив несколько модулей, включая выделение скелета руки, глубокую сверточную сеть для детекции и алгоритм оптимизации на основе народного обучения. Для классификации жестов использовалась 1D CNN, что позволило достичь высокой точно-

сти на видео с динамическими движениями рук, обеспечивая устойчивость результатов в шумных видеопотоках [12]. Пател и соавторы (2023) акцентировали внимание на оптимизации CNN-моделей для работы на мобильных и встроенных устройствах без значительной потери точности – около 90% точности удалось сохранить при уменьшении вычислительных затрат, что крайне важно для внедрения систем распознавания языка жестов в повседневные гаджеты [13].

CNN способны выделять локальные паттерны и текстуры на изображениях, что важно для анализа движений рук. Они также более устойчивы к шуму и легко масштабируются для работы с различными размерами входных данных. Современные исследования показывают высокую точность CNN в сочетании с рекуррентными архитектурами (LSTM, BLSTM, TCN) для анализа динамических жестов [7–13].

Постановка задачи.

Научная проблема:

Современные системы распознавания языка жестов сталкиваются с рядом трудностей: высокая вариативность исполнения жестов, различие ракурсов, освещения, частичное перекрытие пальцев, шумы видеопотока. Существующие методы (классические ИНС, логистическая регрессия, гибридные CNN-LSTM) либо не обеспечивают достаточной точности, либо требуют высокой вычислительной мощности, что затрудняет их применение в реальном времени.

Объект исследования:

Объект исследования – символы русского жестового языка (РЖЯ), формируемые и отображаемые посредством движений рук и пальцев человека.

Предмет исследования:

Алгоритмы и методы машинного обучения, способные анализировать пространственные и временные признаки видеоданных для распознавания жестов. В частности, сверточные нейронные сети (CNN) как инструмент выделения локальных признаков в изображениях и их комбинация с маркерным и безмаркерным подходами.

Цель исследования:

Создание высокоточной системы распознавания языка жестов на базе сверточной нейронной сети, обеспечивающей работу в реальном времени, с возможностью интеграции в системы человеко-машинного взаимодействия.

Задачи исследования:

Для достижения цели необходимо решить следующие задачи:

- ◆ Создание датасета жестов РЖЯ: сбор видеоклипов, аннотирование изображений и разметка символов с различными ракурсами и освещением.
- ◆ Разработка модели сверточной нейронной сети: выбор архитектуры, слоев, функций активации, методов нормализации и регуляризации для эффективного выделения пространственных признаков жестов.
- ◆ Обучение и тестирование модели: подбор оптимальных параметров обучения, контроль за переобучением, оценка точности распознавания на тестовой выборке.
- ◆ Интеграция отслеживания руки с видеопотоком: использование маркеров и алгоритмов компьютерного зрения для устойчивого определения положения кисти и пальцев в реальном времени.
- ◆ Оценка эффективности алгоритма: сравнение с другими методами (ИНС, логистическая регрессия, CNN-LSTM) по критериям точности, скорости и устойчивости к шумам.

Научная гипотеза:

Использование сверточной нейронной сети с интеграцией маркерного отслеживания руки и специализированной архитектуры слоев позволит достичь более высокой точности распознавания жестов ($\geq 95\%$) и обеспечить работу алгоритма в реальном времени, превосходя классические методы и гибридные решения по сочетанию точности и скорости.

Методика исследования. Датасет включал 430 изображений букв РЖЯ (10–15 кадров на каждый жест), обработанных с использованием LabelImg. Данные были разделены на обучающую (80%) и тестовую (20%) выборки.

Модель реализована в Python с использованием Keras и TensorFlow. Архитектура CNN включала:

- ◆ 3 слоя Conv2D с 64 фильтрами, 2 слоя Conv2D с 128 фильтрами (ядро 3×3), ReLU активация.

- ◆ Слои Dropout для предотвращения переобучения.
- ◆ BatchNormalization для нормализации входных данных.
- ◆ MaxPooling для снижения размерности карт признаков.
- ◆ Flatten и Dense слои для классификации жестов.

При подаче изображения или видео на вход система определяет положение руки. После этого процесс разделяется на две ветки:

1) Работа с визуальной частью. Система определяет положение каждого пальца руки, после чего накладывает на неё маркеры. Чтобы процесс не повторялся каждый раз, программа отслеживает руку и обновляет маркеры уже найденных пальцев.

2) Работа с нейросетью. После нахождения руки её изображение отправляется в нейронную сеть, где производится сопоставление с уже известной модели информацией. Если жест совпадает, то выдаётся перевод нужного жеста. Если жест отсутствует в системе, то подбирается наиболее похожий символ.

Пример изображений для буквы С представлен на рис. 1.

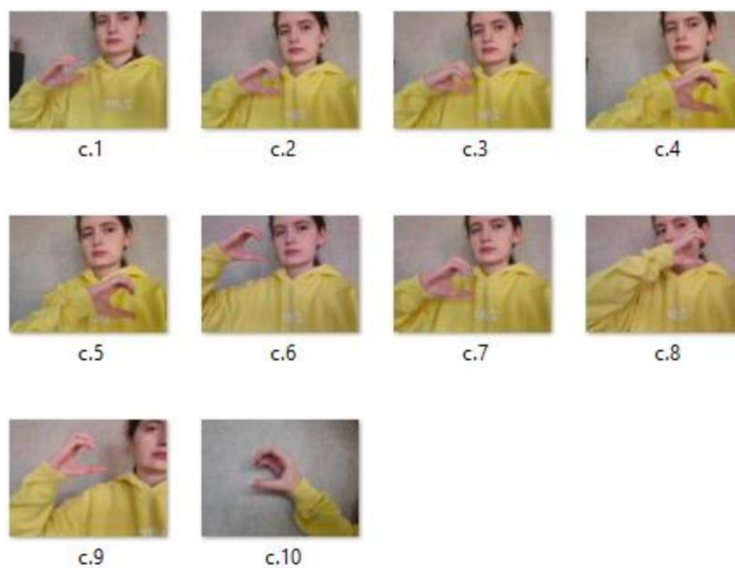


Рис. 1. Набор изображений для буквы С

Для разметки датасета использовался инструмент *LabelImg* – графический инструмент для аннотирования изображений, созданный на языке *Python*. [14].

После разметки данных все изображения были разбиты на два датасета: обучающая и тестовая. Данные были поделены 80% на 20%, то есть 8 изображений каждого жеста в обучающей выборке (344 изображения в сумме) и 2 изображения каждого жеста в тестовой выборке (86 изображений).

Для разработки был выбран язык программирования *Python*. Использовались библиотеки *Keras*, *TensorFlow* и *Scikit-learn*, которые применяются в области машинного обучения. Отдельного внимания заслуживает библиотека *OpenCV* – это библиотека компьютерного зрения с открытым исходным кодом для анализа, классификации и обработки изображений, широко используемая в таких языках, как *C*, *C++*, *Python* и *Java* [15].

Для распознавания жестовых символов использование данной библиотеки является важным. С её помощью создан датасет, который будет записан в режиме реального времени. После обучения модели *OpenCV* сможет определять жесты и предлагать перевод непосредственно в видео.

Модель алгоритма. Для обучения модели распознавания языка жестов была выбрана свёрточная нейронная сеть (*CNN*). Для сравнения применялись три алгоритма, используемые для работы с изображениями: свёрточная нейронная сеть, логистическая регрессия и искусственная нейронная сеть. Исходя из результатов тестирования на тестовом датасете с жестами цифр именно *CNN* подходит лучше всего.

Разработка модели происходила в *Jupyter Notebook*. Обученная модель сохраняется в формате *H5* для дальнейшего использования в видеопотоке. Сам формат *H5* (*Hierarchical Data Format version*) предназначен для хранения и организации больших объёмов данных. Он сохраняет веса модели, обрабатывает сложные иерархические структуры и поддерживает метаданные [16].

Модель в формате *H5* выгружается в скрипт с расширением «*.py*», где происходит распознавание языка жестов в реальном времени из потока данных от видеокамеры. В первую очередь требуется обучить искусственную нейронную сеть распознавать буквы на изображениях и определять их принадлежность к алфавиту. Для этого необходимо разработать функцию и использовать встроенные методы библиотеки *Keras* для создания и обучения модели. Для усовершенствования разработанной модели внедряются дополнительные параметры следующим образом:

1) *model = Sequential()* – Создается объект модели типа *Sequential*, который позволяет последовательно добавлять слои. *Sequential* представляет собой последовательность слоёв, где каждый слой выполняет определённую операцию над входными данными [17]. Этот тип идеально подходит для решения задачи распознавания языка жестов, так как основным принципом алгоритма является классификация и обработка изображений.

2) *Conv2D* – это слой свертки в нейронных сетях, который применяется для обнаружения различных признаков в изображениях. Он используется для свертки входных данных и для извлечения признаков. Каждый фильтр скользит по изображению, применяя операцию свертки, и создает новое изображение (*feature map*) с выделенными признаками [18]. У *Conv2D* есть несколько параметров, включая количество фильтров (*filters*), размер ядра свертки (*kernel_size*), функцию активации (*activation*), способ заполнения (*padding*), шаг (*stride*) и другие. Количество фильтров определяет количество признаков, которые будут извлечены на каждом уровне свертки. Размер ядра свертки определяет область, по которой будет скользить фильтр. Функция активации применяется к выходным данным каждого нейрона. Параметры *padding* и *stride* контролируют размер выходного изображения и шаг, соответственно. Комбинация этих параметров определяет поведение *Conv2D* слоя в нейронной сети.

Для модели создавалось три слоя типа *Conv2D* с 64 фильтрами размером 3x3 и два слоя с 128 фильтрами размером 3x3 со следующими параметрами:

1) *padding = 'Same'* – специальная форма маскировки *padding*, основанная на кодировке последовательностей, установлена в «*Same*». Благодаря этому входные карты признаков будут иметь ту же размерность, что и выходные;

2) *activation = 'relu'* – функция активации *ReLU*;

3) *input_shape = (28,28,1)* – входная форма (28,28,1), что соответствует изображениям размером 28x28 пикселей с одним каналом (например, черно-белое изображение);

4) *Dropout()* - Добавляется слой *Dropout* с вероятностью исключения 0.1, что помогает предотвратить переобучение. В коде таких слоёв несколько с исключением 0.2 и 0.3;

5) *BatchNormalization()* – Добавляется слой *BatchNormalization* для нормализации входных данных каждого мини-пакета. Нормализация пакетов (*batch-normalization*) – метод, способствующий увеличению производительности и устойчивости искусственных нейронных сетей [14]. Основная идея этого метода заключается в том, что некоторые слои нейронной сети получают на вход данные, которые были предварительно обработаны таким образом, чтобы иметь нулевое математическое ожидание и единичную дисперсию;

6) *MaxPool2D(2,2)* – Добавляется слой *MaxPooling* с размером пула 2×2 , который уменьшает размер карт признаков вдвое. Слой *Pooling* в сверточных нейронных сетях используется для уменьшения размерности карты признаков. Он имеет две разновидности: *max-pooling* и *average-pooling*. В данной нейросети используется *max-pooling* для максимального сокращения;

7) *Flatten()* – Карты признаков после последнего слоя *MaxPooling* выравниваются в одномерный вектор. *Flatten* позволяет преобразовать структурированные данные, такие как изображения, в линейный формат, который удобно использовать в нейронных сетях;

8) *Dense(128, activation='relu')* – Добавляется полносвязный слой с 128 нейронами и функцией активации *ReLU (Rectified Linear Unit)*. Это нелинейная функция активации, которая преобразует входное значение в значение от 0 до положительной бесконечности. Если входное значение меньше или равно нулю, то *ReLU* выдаёт ноль, в противном случае – входное значение.

Сама модель принимает вид, изображенный на рис. 2.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 64)	640
dropout (Dropout)	(None, 28, 28, 64)	0
batch_normalization (Batch Normalization)	(None, 28, 28, 64)	256
max_pooling2d (MaxPooling2D)	(None, 14, 14, 64)	0
dropout_1 (Dropout)	(None, 14, 14, 64)	0
conv2d_1 (Conv2D)	(None, 14, 14, 128)	73856
batch_normalization_1 (Batch Normalization)	(None, 14, 14, 128)	512
max_pooling2d_1 (MaxPooling2D)	(None, 7, 7, 128)	0
dropout_2 (Dropout)	(None, 7, 7, 128)	0
conv2d_2 (Conv2D)	(None, 7, 7, 128)	147584
batch_normalization_2 (Batch Normalization)	(None, 7, 7, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 3, 3, 128)	0
flatten (Flatten)	(None, 1152)	0
dense (Dense)	(None, 128)	147584
dropout_3 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 25)	3225

Total params: 374169 (1.43 MB)
Trainable params: 373529 (1.42 MB)
Non-trainable params: 640 (2.50 KB)

Рис. 2. Модель нейронной сети

Модель важно скомпилировать. Это позволяет точно настроить процесс обучения под конкретную задачу и обеспечивает возможность мониторинга и анализа прогресса обучения. Без компиляции модель не сможет корректно обучаться. На данном этапе заканчивается подготовка нейронной сети к её обучению. Указанные слои и параметры позволяют модели достичь высокой точности результатов предсказаний, а компиляция позволит избежать ошибок и переобучения [19].

Для подключения нейросети к видеокамере важно настроить отслеживание руки с помощью точек. Для этого модель была обучена в модуле *mediapipe*:

1) *mp_hands* – предварительно обученную модель для обнаружения и отслеживания рук.

2) *mp_drawing* – предоставляет методы для рисования на изображениях.

3) *mp_drawing_styles* – стили для рисования рук.

Затем создаётся экземпляр класса *Hands* с параметрами *static_image_mode=True* и *min_detection_confidence=0,3*. Эти параметры указывают, что модель будет работать в режиме статического изображения и что минимальная уверенность в обнаружении должна быть 0,3 или выше.

Далее запускается бесконечный цикл, в котором считывается кадр с камеры. Кадр преобразуется из стандартного цветового пространства *BGR* в *RGB*. После этого кадр обрабатывается методом *Hands*, чтобы обнаружить руки. Если метод обнаруживает руки, то для каждой обнаруженной руки рисуются маркеры на кадре. Этот метод принимает следующие аргументы:

- 1) кадр, на котором нужно нарисовать маркеры;
- 2) маркеры, которые были обнаружены методом;
- 3) соединения между маркерами, которые нужно нарисовать;
- 4) стиль для рисования маркеров;
- 5) стиль для рисования соединений.

Были выбраны цветные маркеры, каждый цвет отвечает за отдельный палец: бежевый – большой, фиолетовый – указательный, жёлтый – средний, зелёный – безымянный и синий – мизинец. На каждой фаланге есть маркер, который обозначает место сгиба пальца. Это позволяет отслеживать пальцы в разных ракурсах, рассчитывать расположение скрытых частей. Маркер руки представлен на рис. 3.

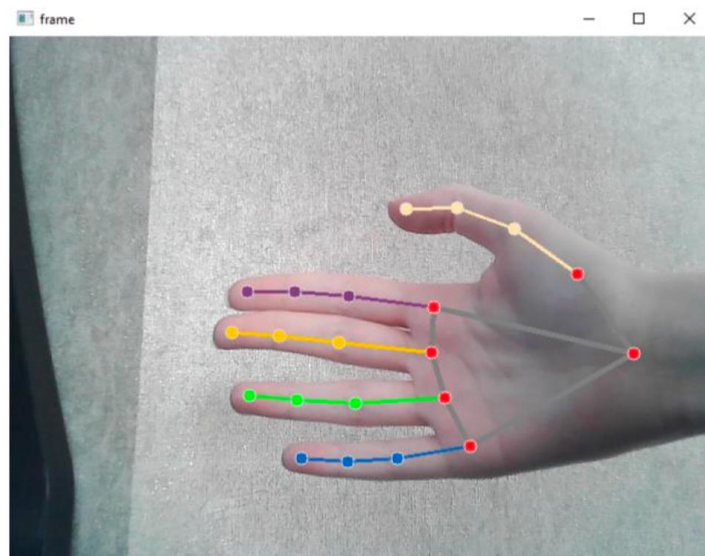


Рис. 3. Маркеры руки

После того как маркеры нарисованы, видео на экране отображается с помощью метода *imshow* библиотеки *OpenCV*. На экране в любом ракурсе отображаются маркеры на руке. Само предсказание будет выводиться в виде чёрной рамки и распознанной буквы над ней.

Результаты. С помощью объявленных параметров обучаем модель на тренировочной выборке. На рис. 4 показаны результаты обучения модели. Процесс занял 28 минут на 10 подходах.

```

history = model.fit(train,label,epochs=10,validation_data=(test,test_label))

Epoch 1/10
/usr/local/lib/python3.10/dist-packages/keras/src/backend.py:5727: UserWarning: ``sparse_categorical_crossentropy`` received ``from_logits=True``,
output, from_logits = _get_logits(
858/858 [=====] - 171s 197ms/step - loss: 0.5237 - accuracy: 0.8366 - val_loss: 0.1583 - val_accuracy: 0.9428
Epoch 2/10
858/858 [=====] - 170s 198ms/step - loss: 0.0405 - accuracy: 0.9865 - val_loss: 0.1695 - val_accuracy: 0.9381
Epoch 3/10
858/858 [=====] - 168s 195ms/step - loss: 0.0291 - accuracy: 0.9897 - val_loss: 0.4449 - val_accuracy: 0.8581
Epoch 4/10
858/858 [=====] - 168s 196ms/step - loss: 0.0285 - accuracy: 0.9905 - val_loss: 0.5009 - val_accuracy: 0.8858
Epoch 5/10
858/858 [=====] - 171s 199ms/step - loss: 0.0189 - accuracy: 0.9944 - val_loss: 0.4315 - val_accuracy: 0.8839
Epoch 6/10
858/858 [=====] - 167s 195ms/step - loss: 0.0216 - accuracy: 0.9930 - val_loss: 0.6361 - val_accuracy: 0.8603
Epoch 7/10
858/858 [=====] - 169s 197ms/step - loss: 0.0197 - accuracy: 0.9937 - val_loss: 0.8175 - val_accuracy: 0.8253
Epoch 8/10
858/858 [=====] - 169s 197ms/step - loss: 0.0198 - accuracy: 0.9941 - val_loss: 0.8957 - val_accuracy: 0.7796
Epoch 9/10
858/858 [=====] - 169s 197ms/step - loss: 0.0195 - accuracy: 0.9941 - val_loss: 0.2757 - val_accuracy: 0.9407
Epoch 10/10
858/858 [=====] - 167s 195ms/step - loss: 0.0047 - accuracy: 0.9985 - val_loss: 0.4476 - val_accuracy: 0.9145

```

Рис. 4. Результаты обучения модели

Модель выдала высокий результат точности на тестовой выборке и составила 99%, что подтверждает высокую эффективность предложенной сверточной нейронной сети (CNN).

Для наглядного сравнения точности предложенного решения с другими методами распознавания жестов представлены в табл. 1.

Таблица 1

Сравнение методов распознавания жестов

Метод	Точность
Сверточная нейронная сеть (CNN)	99%
Классическая ИНС	89%
Логистическая регрессия	84%
CNN-LSTM	96,4%

CNN показала высокую точность (99%), что объясняется способностью сети эффективно выявлять пространственные признаки жестов и устойчивостью к шуму и вариациям освещения.

Классическая ИНС и логистическая регрессия продемонстрировали значительно более низкие показатели (89% и 84% соответственно), так как не учитывают пространственную структуру изображений [20].

CNN-LSTM достигла точности 96,4%, однако требует больше вычислительных ресурсов и сложнее в реализации [9]

Преимущества предложенного алгоритма:

- ◆ Высокая точность и надежность даже при сложных условиях освещения и частичной видимости пальцев.
- ◆ Реальное время обработки: интеграция с OpenCV и MediaPipe позволяет оперативно отображать результаты видеопотока.
- ◆ Сравнение с другими методами демонстрирует, что предложенная CNN превосходит классические ИНС и близка по результату к современным гибридным решениям при меньших вычислительных затратах.
- ◆ Устойчивость к вариациям жестов: система корректно распознает буквы при разных ракурсах, положениях рук и частичном перекрытии пальцев.

Предложенная модель готова к практическому применению в системах человеко-машинного взаимодействия, обеспечивая высокую точность распознавания русского жестового языка и минимальное запаздывание в обработке видеопотока. Дальнейшие работы будут направлены на увеличение объёма и разнообразия обучающих данных, интеграцию временных архитектур для анализа динамических жестов и повышение устойчивости к внешним помехам.

Выводы. Разработан и реализован алгоритм распознавания русского жестового языка (РЖЯ) на базе сверточной нейронной сети (CNN), интегрированной с системой отслеживания рук. Предложенное решение позволяет идентифицировать положение кисти и отдельных пальцев с высокой точностью, обеспечивая корректное распознавание жестов при частичном перекрытии пальцев, различных ракурсах и изменениях освещения.

Проведено обучение и тестирование модели на специализированном датасете, включающем 430 изображений, охватывающих буквы алфавита РЖЯ с различными ракурсами и условиями освещения. Точность распознавания на тестовой выборке составила 99 %, что подтверждает высокую эффективность архитектуры CNN в задачах пространственного анализа изображений жестов.

Практическая значимость исследования заключается в возможности использования предложенного алгоритма для:

- ◆ создания интерактивных систем коммуникации для глухонемых;
- ◆ разработки реабилитационных приложений и образовательных платформ;
- ◆ внедрения в интерфейсы виртуальной и дополненной реальности (VR/AR), где требуется распознавание жестов в реальном времени.

Научная новизна работы заключается в интеграции маркерного отслеживания рук с архитектурой CNN для распознавания жестов, что обеспечивает высокую точность и стабильность распознавания в условиях изменяющейся среды, а также в сравнительном анализе с классическими методами и гибридными CNN-LSTM моделями, подтверждающем преимущество предложенной модели.

Перспективы дальнейших исследований включают:

- ◆ расширение и усложнение датасета за счет включения динамических жестов и словосочетаний;
- ◆ интеграцию временных архитектур (LSTM, TCN) для распознавания динамических жестов и предложений;
- ◆ оптимизацию вычислительных процессов для работы на мобильных и встроенных устройствах;
- ◆ разработку алгоритмов автоматической коррекции ошибок распознавания и повышения устойчивости к шумам и нестандартным положениям руки.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. World Health Organization. Deafness and hearing loss. – 2023. – URL: <https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss> (дата обращения: 10.01.2026).
2. Дialeкты русского жестового языка. – URL: <https://voinfo.ru/vog/2015/05/30/rslidioms/> (дата обращения: 10.02.2024).
3. Cruz M., et al. Sign language recognition: A review // International Journal of Computer Vision. – 2020.
4. Kim H., et al. Marker-based hand gesture recognition systems: A review // Sensors. – 2022.
5. Foteinos K., et al. Visual Hand Gesture Recognition with Deep Learning: A Comprehensive Review of Methods, Datasets, Challenges and Future Research Directions // arXiv preprint arXiv:2507.04465. – 2025.
6. Zhou Y., et al. CNN-LSTM architecture for sign language recognition // IEEE Access. – 2021.
7. Wang L., et al. Spatio-temporal networks for dynamic hand gesture recognition // Neurocomputing. – 2024.
8. Garcia F., et al. Robust CNN models with data augmentation for hand gesture recognition // Pattern Recognition. – 2025.
9. Baihan A., Alutaibi A.I., Alshehri M., et al. Sign language recognition using modified deep learning network and hybrid optimization: a hybrid optimizer (HO) based optimized CNNSa-LSTM approach // Sci Rep. – 2024. – Vol. 14. – Article 26111. – DOI: 10.1038/s41598-024-76174-7.
10. Kumar S., et al. Vision-based markerless hand gesture recognition using deep learning // IEEE Transactions on Multimedia. – 2022.
11. Safa Ameer, Mohamed Ali Mahjoub, Anouar Ben Khalifa. Spatial-temporal generative network based on deep long short-term memory autoencoder for hand skeleton data sequences reconstruction and recognition // Engineering Applications of Artificial Intelligence. – 2025. – Vol. 161. – Part C. – Article 112289.

12. Al Mudawi N., Ansar H., Alazeb A., Aljuaid H., AlQahtani Y., Algarni A., Jalal A., Liu H. Innovative healthcare solutions: robust hand gesture recognition of daily life routines using 1D CNN // *Front. Bioeng. Biotechnol.* – 2024. – Vol. 12. – Article 1401803. – DOI: 10.3389/fbioe.2024.1401803.
13. Patel R., et al. Efficient CNN models for embedded hand gesture recognition // *Journal of Real-Time Image Processing.* – 2023.
14. Буйначев С.К., Боклаг Н.Ю. Основы программирования на языке Python. – Екатеринбург: Изд-во Урал. ун-та, 2014. – 91 с.
15. Жерон О. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow. – М.: Диалектика, 2018. – 688 с.
16. Маккини У. Python и анализ данных: пер. с англ. А.А. Слинкина. – М.: ДМК Пресс, 2020. – 540 с.
17. Сем В. Глубокое обучение: легкая разработка проектов на Python. – СПб.: Питер, 2021. – 352 с.
18. Collette A. H5Py Documentation. Release Sep 06, 2019. – 75 p.
19. Учебник «Машинное зрение». – Т. 2. – URL: <https://copter-space.gitbook.io/uchebnik-mashinnoe-zrenie-tom-2> (дата обращения: 10.04.2025).
20. Esan A., Okomba N., Adebisi T., et al. Comparing CNN and Logistic Regression Results for Automated Detection Tasks. – URL: <https://www.researchgate.net/publication/382880766> (дата обращения: 10.11.2025).

REFERENCES

1. World Health Organization. Deafness and hearing loss, 2023. Available at: <https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss> (accessed 10 January 2026).
2. Dialekty russkogo zhestovogo yazyka [Dialects of russian sign language]. Available at: <https://voginfo.ru/vog/2015/05/30/rslidioms/> (accessed 10 February 2024).
3. Cruz M., et al. Sign language recognition: A review, *International Journal of Computer Vision*, 2020.
4. Kim H., et al. Marker-based hand gesture recognition systems: A review, *Sensors*, 2022.
5. Foteinos K., et al. Visual Hand Gesture Recognition with Deep Learning: A Comprehensive Review of Methods, Datasets, Challenges and Future Research Directions, *arXiv preprint arXiv:2507.04465*, 2025.
6. Zhou Y., et al. CNN-LSTM architecture for sign language recognition, *IEEE Access*, 2021.
7. Wang L., et al. Spatio-temporal networks for dynamic hand gesture recognition, *Neurocomputing*, 2024.
8. Garcia F., et al. Robust CNN models with data augmentation for hand gesture recognition, *Pattern Recognition*, 2025.
9. Baihan A., Alutaibi A.I., Alshehri M., et al. Sign language recognition using modified deep learning network and hybrid optimization: a hybrid optimizer (HO) based optimized CNNSa-LSTM approach, *Sci Rep.*, 2024, Vol. 14, Article 26111. DOI: 10.1038/s41598-024-76174-7.
10. Kumar S., et al. Vision-based markerless hand gesture recognition using deep learning, *IEEE Transactions on Multimedia*, 2022.
11. Safa Ameer, Mohamed Ali Mahjoub, Anouar Ben Khalifa. Spatial-temporal generative network based on deep long short-term memory autoencoder for hand skeleton data sequences reconstruction and recognition, *Engineering Applications of Artificial Intelligence*, 2025, Vol. 161, Part C, Article 112289.
12. Al Mudawi N., Ansar H., Alazeb A., Aljuaid H., AlQahtani Y., Algarni A., Jalal A., Liu H. Innovative healthcare solutions: robust hand gesture recognition of daily life routines using 1D CNN, *Front. Bioeng. Biotechnol.*, 2024, Vol. 12, Article 1401803. DOI: 10.3389/fbioe.2024.1401803.
13. Patel R., et al. Efficient CNN models for embedded hand gesture recognition, *Journal of Real-Time Image Processing*, 2023.
14. Буйначев С.К., Боклаг Н.Ю. Основы программирования на языке Python [Fundamentals of programming in Python]. Екатеринбург: Изд-во Урал. ун-та, 2014, 91 p.
15. Жерон О. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow [Applied machine learning with Scikit-Learn and TensorFlow]. Moscow: Диалектика, 2018, 688 p.
16. Маккини У. Python и анализ данных [Python and data analysis]: transl. from Engl. by A.A. Slinkina. Moscow: DМК Press, 2020, 540 p.
17. Сем В. Глубокое обучение: легкая разработка проектов на Python [Deep learning: easy Python project development]. Saint Petersburg: Питер, 2021, 352 p.
18. Collette A. H5Py Documentation. Release Sep 06, 2019, 75 p.
19. Учебник «Машинное зрение» [Textbook "Machine Vision"]. Vol. 2. Available at: <https://copter-space.gitbook.io/uchebnik-mashinnoe-zrenie-tom-2> (accessed 10 April 2025).
20. Esan A., Okomba N., Adebisi T., et al. Comparing CNN and Logistic Regression Results for Automated Detection Tasks. Available at: <https://www.researchgate.net/publication/382880766> (accessed 10 November 2025).

Бондарева Валерия Евгеньевна – Южно-Российский государственный политехнический университет (НПИ) имени М.И. Платова; e-mail: bondareva.valeria2011@yandex.ru; г. Новочеркасск, Россия; кафедра «Информационные и измерительные системы и технологии»; бакалавр.

Черноморова Татьяна Семеновна – Южно-Российский государственный политехнический университет (НПИ) имени М.И. Платова; e-mail: tatyana.chernomorova@gmail.com; г. Новочеркасск, Россия; кафедра «Информационные и измерительные системы и технологии»; доцент; <https://orcid.org/0009-0003-2457-5263>.

Кривцун Анастасия Витальевна – Южно-Российский государственный политехнический университет (НПИ) имени М.И. Платова; e-mail: anastasia.srstu@gmail.com; г. Новочеркасск, Россия; кафедра «Информационные и измерительные системы и технологии»; доцент; <https://orcid.org/0000-0003-0332-4695>.

Абдулкарем Абеяр – Южно-Российский государственный политехнический университет (НПИ) имени М.И. Платова; e-mail: Abeerabdulsalam15@gmail.com; г. Новочеркасск, Россия; кафедра «Информационные и измерительные системы и технологии»; аспирант; ORCID iD: 0009-0003-1892-9408.

Bondareva Valeria Evgenyevna – Platov South-Russian State Polytechnic University (NPI); e-mail: bondareva.valeria2011@yandex.ru; Novocherkassk, Russia; the Department “Information and Measurement Systems and Technologies”; bachelor.

Chernomorova Tatyana Semyonovna – Platov South-Russian State Polytechnic University (NPI); e-mail: tatyana.chernomorova@gmail.com; Novocherkassk, Russia; the Department “Information and Measurement Systems and Technologies”; associate professor; <https://orcid.org/0009-0003-2457-5263>.

Krivtsun Anastasia Vitalyevna – Platov South-Russian State Polytechnic University (NPI); e-mail: anastasia.srstu@gmail.com; Novocherkassk, Russia; the Department “Information and Measurement Systems and Technologies”; associate professor; <https://orcid.org/0000-0003-0332-4695>.

Abdulkarem Abeer – Platov South-Russian State Polytechnic University (NPI); e-mail: Abeerabdulsalam15@gmail.com; Novocherkassk, Russia; of the Department “Information and Measurement Systems and Technologies,” postgraduate student; ORCID iD: 0009-0003-1892-9408.