

Применение нейросетевых технологий в задачах контроля качества изделий текстильной промышленности

А.А. Казначеева, О.М. Власенко, С.В. Захаркина, М.М. Ступак

*Российский государственный университет имени А.Н. Косыгина
(Технологии. Дизайн. Искусство), г. Москва*

Аннотация: Рассмотрена задача разработки интеллектуальной автоматизированной системы обнаружения дефектов текстильных материалов. Проведен анализ алгоритмов машинного обучения и глубокого обучения применительно к решению задачи контроля качества изделий. Рассмотрена реализация искусственной нейронной сети, реализованной в микрокомпьютере Raspberry Pi и получающей набор входных данных в виде большого потока изображений от высокоскоростной цифровой камеры. Описываются этапы создания модели на языке Python с применением библиотек TensorFlow и Keras. Процесс разработки включает подготовку исходных данных, предназначенных для обучения и тестирования системы, а также проверку работы полученной нейросети, заключающейся в распознавании изображений дефектов на ткани по классификационным признакам.

Ключевые слова: машинное обучение, нейронная сеть, изображения дефектов, текстильный материал, обучение, тестирование, точность.

Создание автоматизированной системы контроля качества изделий текстильной промышленности на основе интеллектуального подхода является сложной и трудоемкой задачей. Решение этой задачи сегодня возможно средствами искусственного интеллекта, с использованием технологии компьютерного зрения, машинного обучения (Machine learning) и нейронных сетей (Deep learning) [1,2].

Задача контроля качества текстильных полотен предполагает обнаружение различных видов дефектов, таких, как пятно, дыра, непрокрас, заломы, нарушение кромки и др., на быстро движущемся материале. Автоматизированная система обнаружения дефектов должна зафиксировать и распознать порок материала, его координаты на рулоне, определить сортность материала согласно нормативным документам. Применение цифровой высокоскоростной камеры для получения изображений материала дает большой массив входных данных для интеллектуальной системы. Задача машинного обучения в этом случае сводится к задаче классификации.

Такой подход позволяет определить дискретный класс для каждого дефекта на основе подготовленных данных [3,4].

Проанализируем методы машинного и глубокого обучения для определения наиболее подходящего алгоритма с целью решения производственной задачи обнаружения дефектов текстильных полотен.

Машинное обучение (МО) – это совокупность методов, которые решают поставленные задачи, используя решения сходных задач с определенными входными данными, обучаются на получаемых данных и развиваются за счет корректировки параметров своих алгоритмов. Особенностью алгоритмов машинного обучения является использование структурированных данных. То есть входной массив данных должен пройти предварительную обработку – фильтрацию, упорядочивание, преобразование в определенный формат, представление в табличной форме, восстановление отсутствующих значимых данных и т.п. Многие алгоритмы машинного обучения работают с численными данными. В случае применения категориальных признаков, их переводят в численный формат [5].

Таким образом, в случае применения машинного обучения для обнаружения дефектов полотен в систему уже должен быть заложен механизм обработки и преобразования данных – изображений, приходящих от камеры. Это целесообразно только при ограниченном количестве обнаруживаемых пороков.

В искусственных нейронных сетях алгоритмы работают аналогично машинному обучению, но входные данные могут быть интерпретированы по-разному. То есть данные на разных уровнях сети определяются и идентифицируются по различным критериям – по аналогии с человеческим мозгом. То есть нейросеть, прогоняя данные через разные уровни, сама в итоге находит те идентификаторы, которые максимально адекватно определяют входные признаки для обучающей модели. Нейронные сети

показывают хороший результат в случае большого объема входных неструктурированных данных.

Проанализируем массив получаемых данных в системе контроля качества текстильных материалов на разбраковочных машинах. Скорость перемотки материалов на типовых разбраковочных машинах варьируется от 15 до 38 м/мин (0.25 – 0.63 м/с). Длина рулона ткани может составлять от 40 до 100 м, трикотажного полотна 70-90 м. В случае применения цифровой камеры со скоростью съемки 30 кадров/с на вход микроконтроллера будет поступать от 4800 до 12000 изображений на один рулон, для камеры с FPS 12 от 1920 до 4800. В соответствии с ГОСТ 161-86, сортность ткани определяется по 28 местным и 11 распространенным порокам. 20 пороков тканей недопустимы и подлежат вырезу.

Таким образом, в случае постановки задачи обнаружения максимального количества пороков согласно нормативным документам, количество классов в задаче классификации велико, что при большом количестве входных кадров приводит к необходимости использовать нейронную сеть [4].

Помимо подготовки данных машинное обучение включает в себя этапы построения модели – определение задачи, подбор алгоритмов; оценку модели (циклический скользящий контроль, замеры производительности и регулировка параметров); оптимизацию, масштабирование и прогнозирование. Относительно применения для прикладной задачи обнаружения дефектов текстильных полотен рассмотрим ряд алгоритмов классификации, основанных на технологии обучения с учителем. В этом случае целевая переменная доступна в обучающей выборке [5].

Алгоритм логистической регрессии является линейным. В его основе лежит анализ количественных признаков входных данных и в зависимости от них происходит разнесение данных по категориям [5]. При этом

логистическая регрессия оперирует понятием вероятности целевых классов. Этот простой алгоритм машинного обучения хорошо подходит для решения задачи двоичной классификации, когда необходимо разложить данные по двум категориям. В этом случае логистическая регрессия строится на основе сигмоиды, которая, вне зависимости от переданного ей входного значения, всегда возвращает значение в промежутке между 0 и 1:

$$f(x) = \frac{1}{1+e^{-x}}. \quad (1)$$

Для построения классификатора в алгоритме логистической регрессии определяют границу, которая разбивает входные данные по целевым классам (категориям). Количество параметров модели, которые необходимо определить, обусловлено количеством измерений, в которых описывается разрешающая граница. На плоскости, в двух измерениях – это линия. Границей разделения для сигмоиды, как правило, принимается значение 0.5.

Для оценки точности модели используется статистическое отклонение или строится матрица ошибок, которая содержит истинно положительные, истинно отрицательные, ложно положительные и ложно отрицательные показатели [5]. Далее оценивают зависимости между данными показателями. Для оптимизации модели используется градиентный спуск. Алгоритм логистической регрессии является простым и удобным для расчета, хорошо работает с большим набором данных, но при этом его производительность снижается при увеличении степени нелинейности разрешающей границы. Еще одним недостатком алгоритма является возможность переобучения модели.

Метод опорных векторов (МОВ) несмотря на свою линейную природу относится к классу нелинейных алгоритмов, благодаря применению нелинейной функции ядра [6]. Ядро представляет собой совокупность математических уравнений, описывающих дополнительное пространство с

данными. В данном пространстве алгоритм определяет линейную границу, которая после возвращения в обычное пространство оказывается нелинейной. Алгоритм определяет линию или гиперплоскость в случае большего числа измерений, оптимально разделяющую классы объектов, и вычисляет максимальный зазор между точками по обе стороны от линии решений.

Основным недостатком нелинейных алгоритмов является потребность в больших вычислительных ресурсах и проблема масштабируемости к большому объему данных. В качестве оценки принимается вероятность ошибки обобщения. В качестве уравнения оптимальной гиперплоскости можно использовать соотношение [6]:

$$(\bar{w}_0 \cdot \bar{x}) - \frac{c_1(\bar{w}_0) + c_2(\bar{w}_0)}{2} = 0, \quad (2)$$

где \bar{w}_0 – нормирующий вектор; \bar{x} – вектор входных признаков; $c_1(\bar{w}_0)$ и $c_2(\bar{w}_0)$ – параметры разделяющей гиперплоскости:

$$c_1(\bar{w}_0) = \min_{y_i=1}((\bar{w} \cdot \bar{x}_i)), \quad (3)$$

$$c_2(\bar{w}_0) = \max_{y_i=1}((\bar{w} \cdot \bar{x}_i)).$$

Расстояние от ближайших точек положительной и отрицательной частей выборки до оптимальной гиперплоскости – геометрический край ошибки, определяется по формуле:

$$\rho(\bar{w}_0^*) = \frac{1}{\|\bar{w}_0\|}. \quad (4)$$

Ошибка обобщения определяется из соотношения, связывающего число опорных векторов d ($d > 2$) и размер выборки l :

$$\varepsilon \leq \frac{d \ln l}{l - d}. \quad (5)$$

Метод k -ближайших соседей является нелинейным алгоритмом со

встроенной поддержкой многоклассовых задач [7]. По данному методу новые данные разносят по классам путем их сравнения с записями из обучающей выборки. То есть анализируемый объект относится к тому классу, к которому принадлежат большинство его k соседей из обучающей выборки. Обычно используется 3, 5 или 9 соседей и выбирается чаще других встречающийся класс. Выбор количества соседей имеет важное значение. Маленькое значение k позволит создать модель, точно дающую прогноз на обучающем множестве данных. Но при этом такая модель будет иметь низкую точность прогноза для тестовых данных. С другой стороны, высокое значение показателя k чрезмерно усложняет модель.

В случае, когда количество классов, по которым надо разнести данные, больше 3, целесообразно использовать метод взвешенных ближайших соседей [5]. В данном методе каждому i -му соседу присваивается определенный вес, который зависит от расстояния до соседа. При этом в случае уменьшения расстояния, вес будет уменьшаться. Объект относят к тому или иному классу по наибольшему суммарному весу среди k ближайших соседей. При использовании данного алгоритма важно нормировать значения признаков, иначе один признак с максимальным значением может стать преобладающим, а признаки с маленькими значениями не будут учитываться при классификации.

В качестве метрик расстояния до k соседей применяются Евклидова метрика, расстояние Чебышева, Манхэттенское расстояние, Метрика Минковского и др. При использовании Евклидовой метрики расстояние между соседями d для набора числовых данных n_i можно определить по формуле [5]:

$$d = \sqrt{n_1^2 + n_2^2 + \dots + n_n^2} \quad (6)$$

Предсказание для новых точек определяется по ближайшей известной точке, предполагая, что эта точка принадлежит к этому же классу. Если

рассматривается всего одна ближайшая запись, то классификатор работает по одному ближайшему соседу.

Достоинство метода k -ближайших соседей заключается в работе с любым количеством категорий в задаче классификации, а также в хорошей масштабируемости. При этом модель имеет всего два параметра: количество соседей k и метрику расстояния. К недостаткам метода можно отнести большие вычислительные затраты, а также сложность работы с категориальными данными. Метод k -ближайших соседей хорошо работает в случае, когда обучение необходимо провести быстро, а само предсказание по модели происходит медленнее.

Деревья решений – высокоточный нелинейный алгоритм, широко применяемый для решения задач классификации. Алгоритм определяет по обучающей выборке «самые важные» переменные и помещает их на верх дерева, постепенно переходя затем к менее важным переменным [6]. Одним из преимуществ данного алгоритма является нечувствительность к несущественным признакам, а также наборам данных с пропущенными значениями.

Алгоритм машинного обучения, известный под названием Бустинг, используется для уменьшения количества ошибок при прогнозном анализе данных. В результате последовательного объединения нескольких деревьев решений алгоритм строит ансамбли моделей. Каждому дереву решений присваиваются определенные веса в зависимости от результата. Неправильным классификациям из первого дерева решений присваивается больший вес, после чего данные передаются в следующее дерево. После прохождения всех деревьев бустинг объединяет слабые классификаторы в общий алгоритм прогнозирования [7].

Анализируя вышеперечисленные алгоритмы машинного обучения применительно к решению задачи обнаружения дефектов на движущихся

текстильных полотнах, приходим к выводу, что они применимы только в случае ограниченного количества определяемых пороков в узком диапазоне их вариаций. Для полноценной интеллектуальной системы контроля качества материала, определения его сортности, необходимо применение алгоритма, который бы решал задачу классификации с учетом нечеткой структурированности входных данных. Такой алгоритм реализует искусственная нейронная сеть.

Для оценки и прогнозирования значений на основе большого количества входных данных целесообразно использовать нейронную сеть на основе метода обратного распространения ошибки. Несмотря на достаточно высокий по сравнению с другими уровень сложности, такой подход в определенных случаях позволяет получить хорошие результаты, и используется для многих методов глубокого обучения [8-10].

В общем случае нейронная сеть – это компьютерная программа, в основе которой лежат определенные алгоритмы анализа и обработки данных. Многие специалисты проводят аналогию между работой нейронной сети и человеческим мозгом. Существуют разные варианты архитектуры нейронных сетей, каждая из которых хорошо работает со своими входными данными. Например, в решении задач обработки изображений хорошо показывает себя сверточная нейронная сеть, впервые предложенная Я. Лекуном [8]. Такие сети распознают и классифицируют образы любого разрешения, хорошо масштабируются. В данной работе для решения задачи контроля качества текстильных изделий, используется сверточная нейронная сеть.

Чтобы распознать изображение, нейронная сеть должна обучиться на некоторых наборах входных данных. Каждый объект обучающей выборки принадлежит к одному из k классов $(0, 1, 2, \dots, k-1)$, которые пронумерованы. После обучения, полученная модель должна классифицировать неизвестные ей изображения из тестовых наборов, то есть указывать, к какому классу

относится данный объект [11].

В данной работе приведена реализация нейронной сети для оценки качества текстильных материалов. Аппаратную основу системы распознавания составляют микрокомпьютер Raspberry Pi и видеочамера Raspberry Pi Camera V1.3. Нейронная сеть разработана на языке Python в среде Google Colab с использованием библиотек TensorFlow и Keras. Изображения с чамеры поступали на микрокомпьютер для обработки нейронной сетью, задачей которой являлся анализ и определение дефектов.

Нейросети требуют большого количества и качества данных. Для задачи распознавания дефектов на ткани в рассматриваемом случае набор данных для нейросети создавался вручную. Данные были структурированы как дерево из папок. В каждую папку помещались изображения с дефектами. Для отработки алгоритма были выбраны такие дефекты, как пятно и дыра.

Одним из альтернативных вариантов создания нейронной сети в задачах контроля качества текстильных изделий является ее реализация в программной среде Matlab с применением модуля Deep Learning Toolbox. Этот модуль поддерживает разработку приложений для машинного обучения, а также приложения для сверточных нейронных сетей [12]. Была проведена разработка и настройка простой сети глубокого обучения для задачи классификации. В результате проведенных исследований, обучения и тестирования, точность работы нейронной сети составила 60%. Такая точность является недостаточной для применения в технологическом процессе, поэтому требуется проведение процедуры ее улучшения и оптимизации.

Другим инструментом создания нейронной сети является язык программирования высокого уровня Python. К преимуществам Python относят простоту и многофункциональность, возможность загрузки программного кода в микрокомпьютер Raspberry Pi, а также наличие

библиотек, подходящих для разработки искомой модели [13,14]. Например, библиотеки TensorFlow и Keras [15,16], которые используют открытый исходный код и позволяют производить числовые расчеты с применением блок-схем данных. Благодаря своей простоте и удобству, эти библиотеки активно используются для разработки моделей глубокого обучения.

Как уже было сказано выше, разработка нейронной сети проводилась в интерактивной облачной среде Google Colab. В основу данного сервиса положен инструментарий Jupyter Notebook для работы на Python. На первом этапе разработки необходимо установить модули библиотек, и загрузить в проект заранее подготовленный архив с данными. В наборе данных содержатся 2 класса дефектов: пятно и дыра [10, 16]. Всего входной набор составляет 2000 изображений дефектов.

Для загрузки набора изображений в Tensorflow используется утилита `image_dataset_from_directory`. Эта утилита из каталога изображений формирует файл `tf.data.Dataset`. В данном каталоге изображения одного класса находятся в отдельной папке [11]. На втором этапе формируется каталог, в котором создается новый набор данных.

Каталог изображений был разбит на две части: обучающую выборку из 1800 изображений и тестовую выборку из 200 изображений.

При использовании утилиты `image_dataset_from_directory`, вводится параметр `validation_split`, который отвечает за формирование поднабора, отвечающего за проверку работы нейронной сети. Количество изображений, входящих в поднабор, составляет 10% от общего количества данных, используемых для обучения. Размер каждого изображения - 277×277 пикселей. Далее формируется список с названиями классов, а также выводится несколько изображений из набора данных «Дефекты» (рис. 1).

На третьем этапе, определяется нейронная сеть для распознавания дефектов. На выходе формируется 2 нейрона по количеству классов в наборе

данных.

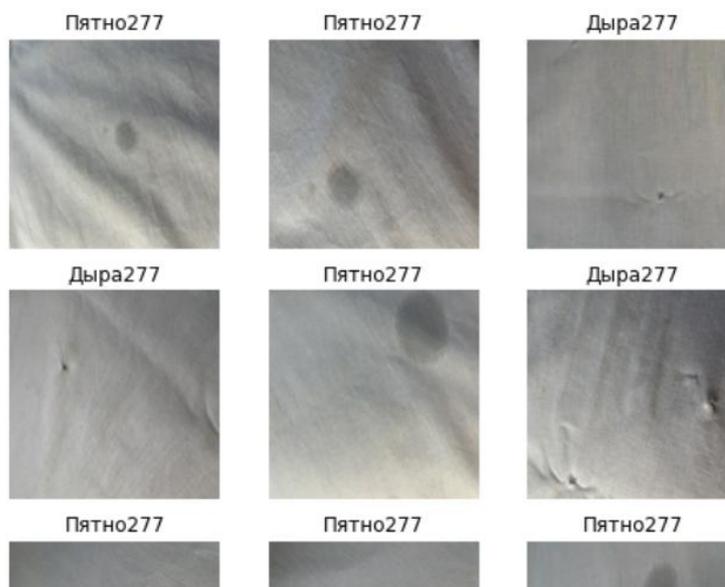


Рис. 1. – Вывод изображений дефектов

На рис. 2 (а, б) показаны графики качества обучения нейронной сети.

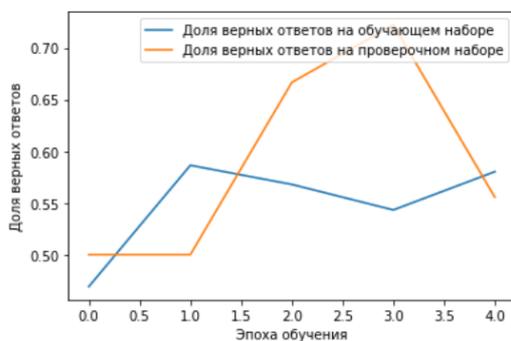


Рис. 2 (а) – Графики верных ответов

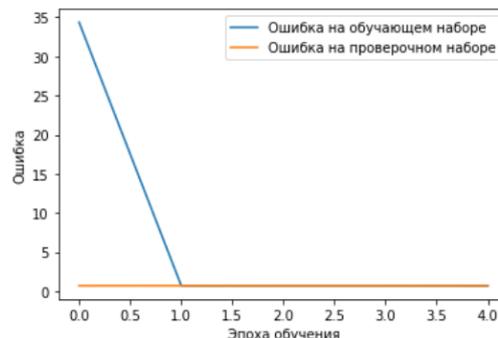


Рис. 2 (б) – Графики ошибок

Результатом проверки работы полученной модели является распознавание изображения дефекта по классификационным признакам. Для этого необходимо:

- загрузить программу в виде файла формата Hierarchical Data Format (HDF);
- добавить данное изображение в модель.

После загрузки изображения выполняется команда Image (рис. 3). В

качестве параметров указываются размеры изображения в пикселях.

```
[74] Image(img_path, width=277, height=277)
```



Рис. 3 – Вывод загруженного изображения дефекта на экран

На рис. 4 показан результат работы обученной нейросети. На выходе формируется два нейрона по количеству классов. То есть, из двух имеющихся классов наиболее вероятно, изображение относится к классу с номером 0 и с названием «Дыра277». Для печати номера и названия класса применяется метод `np.argmax`.

```
▶ prediction = np.argmax(prediction)
print("Номер класса:", prediction)
print("Название класса:", classes[prediction])
```

Номер класса: 0
Название класса: Дыра277

Рис. 4 – Вывод на экран номера и названия класса

Таким образом, в случае применения для реализации нейронной сети для задачи контроля качества текстильных изделий модуля Deep Learning Toolbox и приложения для сверточных нейронных сетей программной среды

Matlab, использование алгоритма классификации дало точность работы нейронной сети 60%. Используемая модель требует проведение процедуры ее улучшения и оптимизации.

Среда разработки Python с встроенными библиотеками TensorFlow и Keras является более мощным средством машинного обучения и разработки нейронные сети. Важным достоинством данного инструмента является возможность загрузки программного кода в микрокомпьютер Raspberry Pi.

В данной статье описан процесс создания и настройки нейронной сети, ее обучение и тестирование. Получены следующие результаты ее работы: количество эпох – 5; время обучения одной эпохи примерно 50 секунд; максимальная доля верных ответов – 72%. Результатом проверки работы полученной модели является распознавание изображения дефекта по классификационным признакам с долей верных ответов на тестовых данных 75%.

Таким образом, применительно к решению задачи обнаружения различных дефектов на движущихся текстильных полотнах наиболее мощным и эффективным средством являются нейронные сети, а рациональным программным средством создания нейросети – среда разработки Python с встроенные библиотеки для глубокого машинного обучения TensorFlow и Keras. Данная среда программирования является универсальной, может работать на разных аппаратных платформах и операционных системах, а также позволяет загружать программный код непосредственно в микроконтроллер, что является значимым фактором для использования ее в технологических автоматизированных системах оценки качества текстильных материалов.

Литература

1. Chao Li, Jun Li, Yafei Li, Lingmin He, Xiaokang Fu and Jingjing Chen. Fabric Defect Detection in Textile Manufacturing: A Survey of the State of the Art

// Security and Communication Networks. Volume 2021, Article ID 9948808.

2. Aqsa Rashe, Bushra Zafar, Amina Rasheed, Nouman Ali. Fabric Defect Detection Using Computer Vision Techniques: A Comprehensive Review // Mathematical Problems in Engineering Volume 2020, Article ID 8189403.

3. Yundong Li, Cheng Zhang. Automated vision system for fabric defect inspection using Gabor filters and PCNN // Li and Zhang SpringerPlus (2016) 5:765 DOI 10.1186/s40064-016-2452-6.

4. Tamnun E Mursalin, Fajrana Zebin Eishita, Ahmed Ridwanul Islam. Fabric defect inspection system using neural network and microcontroller // Journal of Theoretical and Applied Information Technology. 2008. P. 560 – 570.

5. Brink Hendrik, Richards Joseph W., Fetherolf Mark. Real – World Machine Learning. Shelter Island, NY: Manning Publications Co. 2017. 298 p.

6. Вьюгин В.В. Математические основы машинного обучения и прогнозирования. // МЦНМО: М. 2014. 304 с.

7. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. Data Mining, Inference, and Prediction, Second Edition // Springer Science, LLS 2009. New York. USA. DOI: 10.1007/b94608. 737 p.

8. Николенко С., Кадури А., Архангельская Е. Глубокое обучение. // СПб.: Питер. 2018. 480 с.

9. Крейман Г. Биологическое и компьютерное зрение. // ДМК-Пресс: М. 2022. 314 с.

10. Тарик Рашид. Создаем нейронную сеть. // ООО «Диалектика»: СПб. 2019. 272 с.

11. Казначеева А.А., Ступак М.М. Разработка базы знаний специалиста предметной области контроля качества текстильных материалов. // Международная научно-техническая конференция «Дизайн, технологии и инновации в текстильной и легкой промышленности (ИННОВАЦИИ-2022). Часть 2. М.: ФГБОУ ВО «РГУ им. А.Н. Косыгина». 2022. С. 249-253.



12. Гилат Амос. MATLAB. Теория и практика. // ДМК-Пресс: М. 2016. 416 с.
13. Казначеева А.А., Ступак М.М. Обзор готовых библиотек реализации нейронной сети в алгоритмах автоматизированного обнаружения дефектов на ткани. // Сборник научных трудов кафедры автоматизики и промышленной электроники Российского государственного университета им. А.Н. Косыгина. М. 2023. С. 34-37.
14. Казначеева А.А., Захаркина С.В., Власенко О.М., Рыжкова Е.А. Разработка автоматизированной системы обнаружения дефектов на ткани с применением компьютерного зрения. // Инженерный вестник Дона. 2021. №12. URL: ivdon.ru/ru/magazine/archive/n12y2021/7339
15. Клетте Р. Компьютерное зрение. Теория и алгоритмы. // ДМК-Пресс: М. 2019. 506 с.
16. Солем Я. Программирование компьютерного зрения на языке Python. // ДМК-Пресс: М. 2016. 311 с.

References

1. Chao Li, Jun Li, Yafei Li, Lingmin He, Xiaokang Fu and Jingjing Chen. 2021. Article ID 9948808.
 2. Aqsa Rashe, Bushra Zafar, Amina Rasheed, Nouman Ali. 2020. Article ID 8189403.
 3. Yundong Li, Cheng Zhang. 2016. 5:765 DOI 10.1186/s40064-016-2452-6.
 4. Tamnun E Mursalin, Fajrana Zebin Eishita, Ahmed Ridwanul Islam. Journal of Theoretical and Applied Information Technology. 2008. P. 560 – 570.
 5. Brink Hendrik, Richards Joseph W., Fetherolf Mark. Real -World Machine Learning. Shelter Island, NY: Manning Publications Co. 2017. 298 p.
 6. V'yugin V.V. Matematicheskie osnovy mashinnogo obucheniya i prognozirovaniya [Mathematical foundations of machine learning and forecasting]. MTsNMO: M. 2014. 304 p.
-

7. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. Data Mining, Inference, and Prediction, Second Edition. Springer Science, LLS 2009. New York, USA. DOI: 10.1007/b94608. 737 p.

8. Nikolenko S., Kadurin A., Arkhangel'skaya E. Glubokoe obuchenie [Deep learning]. SPb.: Piter. 2018. 480 p.

9. Kreyman G. Biologicheskoe i komp'yuternoe zrenie [Biological and computer vision]. Moskva. DMK-Press. 2022. 314 p.

10. Tariq R. Sozdaem neyronnyuyu set' [Creating a neural network]. Sankt-Peterburg: OOO «Dialektika». 2019. 272 p.

11. Kaznacheeva A.A., Stupak M.M. Dizayn, tekhnologii i innovatsii v tekstil'noy i legkoy promyshlennosti (INNOVATSII-2022): sbornik materialov Mezhdunarodnoy nauchno-tekhnicheskoy konferentsii. Chast' 2 (International scientific and technical conference “Design, technology and innovation in the textile and light industry (INNOVATIONS-2022). Part 2). M.: FGBOU VO «RGU im. A.N. Kosygina». 2022. pp. 249-253.

12. Gilat Amos. MATLAB. Teoriya i praktika [MATLAB. Theory and practice]. Moskva. DMK-Press. 2016. 416 p.

13. Kaznacheeva A.A., Stupak M.M. Sbornik nauchnykh trudov kafedry avtomatiki i promyshlennoy elektroniki Rossiyskogo gosudarstvennogo universiteta im. A.N. Kosygina (Collection of scientific works of the Department of Automation and Industrial Electronics of the Russian State University. A.N. Kosygina). M. 2023. pp. 34-37.

14. Kaznacheeva A.A., Zakharkina S.V., Vlasenko O.M., Ryzhkova E.A. Inzhenernyj vestnik Dona. 2021. №12. URL: ivdon.ru/ru/magazine/archive/n12y2021/7339

15. Clette R. Komp'yuternoe zrenie. Teoriya i algoritmy` [Computer vision. Theory and algorithms]. DMK-Press: M. 2019. 506 p.

16. Yan Solem. Komp'yuternoe zrenie. Teoriya i algoritmy` [Computer Vision



Programming in Python]. DMK-Press: М. 2016. 312 p.

Дата поступления: 5.12.2023

Дата публикации: 11.01.2024