



Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. 2024. Т. 24, вып. 3. С. 432–441
Izvestiya of Saratov University. Mathematics. Mechanics. Informatics, 2024, vol. 24, iss. 3, pp. 432–441
<https://mmi.sgu.ru> <https://doi.org/10.18500/1816-9791-2024-24-3-432-441>, EDN: NIUIGP

Научная статья
УДК 004.932.2

Выделение признаков на изображениях на основе интегральных преобразований при решении задач классификации фрагментов фотоснимков

А. А. Клячин

Волгоградский государственный университет, Россия, 400062, г. Волгоград, Университетский пр-т, д. 100

Клячин Алексей Александрович, доктор физико-математических наук, заведующий кафедрой математического анализа и теории функций, aleksey.klyachin@volsu.ru, <https://orcid.org/0000-0003-3293-9066>, AuthorID: 9530

Аннотация. В статье предложен способ вычисления признаков на изображении для формирования обучающего набора данных при решении различных задач классификации видеоснимков. Данный метод предполагает использование известных интегральных преобразований — преобразование Радона и функции Стеклова. Приводится сравнение предложенного способа со сверточными нейронными сетями как по проценту правильного предсказания, так и по времени его выполнения. В качестве тестовой задачи рассмотрена задача определения наличия изображения автомобиля на фотоснимке.

Ключевые слова: преобразование Радона, функции Стеклова, сверточная нейронная сеть, методы машинного обучения

Для цитирования: Клячин А. А. Выделение признаков на изображениях на основе интегральных преобразований при решении задач классификации фрагментов фотоснимков // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. 2024. Т. 24, вып. 3. С. 432–441. <https://doi.org/10.18500/1816-9791-2024-24-3-432-441>, EDN: NIUIGP

Статья опубликована на условиях лицензии Creative Commons Attribution 4.0 International (CC-BY 4.0)

Article

Extraction of features in images based on integral transformations in solving problems of classification of fragments of photographs

A. A. Klyachin

Volgograd State University, 100 Prosp. Universitetsky, 400062 Volgograd, Russia

Alexey A. Klyachin, aleksey.klyachin@volsu.ru, <https://orcid.org/0000-0003-3293-9066>, AuthorID: 9530

Abstract. The article proposes a method for calculating features in an image to form a training data set for solving various problems of video image classification. This method involves the use of well-known integral transformations — the Radon transform and the Steklov function. The proposed method is compared with convolutional neural networks both in terms of the percentage of correct prediction and in terms of its execution time. As a test task, the problem of finding a fragment of a photograph containing an image of a car is considered.

Keywords: Radon transform, Steklov functions, convolutional neural network, machine learning methods

For citation: Klyachin A. A. Extraction of features in images based on integral transformations in solving problems of classification of fragments of photographs. *Izvestiya of Saratov University. Mathematics.*



Mechanics. Informatics, 2024, vol. 24, iss. 3, pp. 432–441 (in Russian). <https://doi.org/10.18500/1816-9791-2024-24-3-432-441>, EDN: NIUIGP

This is an open access article distributed under the terms of Creative Commons Attribution 4.0 International License (CC-BY 4.0)

Введение

В современном мире происходит стремительное возрастание использования видеoinформации как в повседневной жизни людей, так и в различных системах мониторинга, технического зрения, регистрации и передачи больших объемов видеоданных. В частности, методы цифровой обработки изображений применяются для предварительного анализа изображений с последующим решением задач компьютерного зрения (поиск и распознавание объектов на изображении, 3D реконструкция объектов и т. д.). Компьютерное зрение — область науки, которая занимается задачами, связанными с анализом изображений и видео. К таким задачам относятся: распознавание текста; биометрия (для распознавания людей используются изображения лица, радужной оболочки глаз, отпечатки пальцев); видеоаналитика (видеорегистрация движения автомобилей или отслеживания потоков людей, детектирования различных нестандартных ситуаций); анализ спутниковых снимков; 3D-анализ (реконструкция объемных моделей); анализ информации, получаемой с датчиков, для систем предотвращения ДТП и все более сложных автопилотов и т. п. Методы компьютерного зрения используются для решения задач, которые условно можно разделить на низкоуровневые и высокоуровневые. Высокоуровневые задачи отвечают на вопросы, какой объект изображен на картинке, к какому классу он относится. Для решения этих задач чаще всего используются методы машинного обучения. При решении низкоуровневых задач производятся различные манипуляции непосредственно с уровнями яркости точек (пикселей). Чаще всего низкоуровневые задачи используются как составная часть более сложных задач распознавания. Например, предобработка картинки позволяет алгоритмам машинного обучения лучше понять, что на ней изображено.

В настоящей работе предложен способ выделения признаков на изображении, основанный на применении интегральных преобразований. Некоторые свойства таких преобразований позволяют представить входное изображение некоторым набором числовых значений, по которым можно вести их классификацию. Для разработки кода и тестирования предложенного нами метода мы воспользовались соответствующим программным обеспечением. На сегодняшний день, пожалуй, самая популярная библиотека для решения задач низкоуровневого компьютерного зрения — это `OpenCV`. В ней содержится огромное количество алгоритмов, есть интерфейсы для многих языков программирования, в том числе `C++` и `Python`. Помимо этого, мы использовали уже готовые реализации алгоритмов машинного обучения в виде соответствующих модулей языка `Python`:

1) `Scikit-learn` — это один из наиболее широко используемых пакетов `Python` для `Data Science` и `Machine Learning`;

2) `Keras` — это библиотека глубокого обучения, представляющая собой высокоуровневый API, написанный на `Python` и способный работать поверх `TensorFlow`, `Theano` или `CNTK`.

1. Общая идея метода

При решении задач компьютерного зрения, как правило, используют два подхода. Первый основан на заранее известной информации об объектах, которые изображены на снимках: это может быть их форма (используется контурный анализ), цветовой тон (используются различные методы цветовой фильтрации) и т. д. В этом случае для выделения и распознавания объектов применяются базовые алгоритмы обработки изображений [1]. Вторым подходом основан на применении методов машинного обучения, что предполагает выделение некоторых признаков, характеризующих изображенные объекты (см., например, [2–10]). Одним из универсальных способов получить такие признаки является использование сверточной нейронной сети. Обусловлено это тем, что сверточная нейронная сеть содержит слои, которые применяют



фильтры для выделения нужных признаков на изображении. Процесс обучения на достаточно большом наборе входных изображений позволяет определить наиболее характерные признаки, присутствующие в данном наборе. В следующих работах можно познакомиться как с задачами, решаемыми при помощи сверточных нейронных сетей, так и с их архитектурой [11–18]. В настоящей работе представлен иной подход вычисления признаков, основанный на различных интегральных преобразованиях. В частности, применяются идеи, опирающиеся на два интегральных преобразования: преобразование Радона [19] и функции Стеклова [20].

Преобразование Радона — интегральное преобразование функции многих переменных, родственное преобразованию Фурье. Впервые введено в 1917 г. австрийским математиком Иоганном Радонам. Отметим, что преобразование Радона обладает свойством обратимости, т. е. возможностью восстанавливать исходную функцию по ее преобразованию Радона.

Для произвольной функции $z = f(x, y)$, определенной на плоскости (x, y) , полагаем

$$R(r, \varphi) = \int_{-\infty}^{+\infty} f(r \cos \varphi + t \sin \varphi, r \sin \varphi - t \cos \varphi) dt,$$

где $r > 0$ и $\varphi \in [0, 2\pi)$. Полученная функция $R(r, \varphi)$ есть интеграл по всем прямым плоскости, которые описываются уравнениями

$$x = r \cos \varphi + t \sin \varphi, \quad y = r \sin \varphi - t \cos \varphi, \quad t \in (-\infty, +\infty).$$

Данная функция $R(r, \varphi)$ называется *преобразованием Радона функции $f(x, y)$* .

Если $f(x, y)$ задает яркость точки (x, y) на изображении, то вместо интегралов по всем прямым будем рассматривать среднюю яркость по всем вертикальным, горизонтальным и диагональным линиям, т. е. вычислять среднюю яркость по строкам, столбцам и диагоналям изображения. Понятно, что в этом случае мы теряем часть информации. Однако это вполне допустимо, так как при решении задач классификации изображений нет необходимости в их полном восстановлении.

Функции Стеклова введены русским математиком В. А. Стекловым (1907 г.) для решения задач, связанных с представлением функций в виде рядов по системам собственных функций. Для заданной функции $z = f(x, y)$ и $h > 0$ полагают

$$f_h(x, y) = \frac{1}{h^2} \int_{x-h/2}^{x+h/2} \int_{y-h/2}^{y+h/2} f(u, v) dv du.$$

Таким образом, $f_h(x, y)$ есть среднее значение функции $f(x, y)$ в квадрате $[x - \frac{h}{2}, x + \frac{h}{2}] \times [y - \frac{h}{2}, y + \frac{h}{2}]$. Известно, что для непрерывной функции $f(x, y)$ семейство функций $f_h(x, y)$ равномерно сходится к исходной функции на компактных множествах при $h \rightarrow 0$.

Для случая изображений мы заменим интегралы на суммы, и результат такого преобразования будем рассматривать в отдельных точках. В итоге получим следующую конструкцию. Разбиваем входное изображение на блоки пикселей и в каждом блоке вычисляем среднее значение яркости. Полученный набор значений записываем в отдельный массив. Здесь также происходит потеря информации, и восстановить по этим данным изображение невозможно. Однако эти потери несущественны при решении задач распознавания объектов на изображении, что будет показано далее.

2. Выделение признаков с применением интегральных преобразований

Любое изображение размера $n \times m$ в оттенках серого цвета можно считать функцией двух переменных $f(x, y)$ как функцию яркости точки (x, y) , где x — номер столбца, y — номер



строки. Введем обозначения

$$M_y^f(x) = \frac{1}{m} \cdot \sum_{y=0}^{m-1} f(x, y), \quad M_x^f(y) = \frac{1}{n} \cdot \sum_{x=0}^{n-1} f(x, y). \quad (1)$$

Другими словами, $M_y^f(x)$ — средняя яркость пикселей изображения f , расположенных в столбце x , $M_x^f(y)$ — средняя яркость пикселей изображения f , расположенных в строке y .

Помимо этого, рассмотрим средние яркости, вычисленные по диагональным направлениям. Предположим, что $m = n$ (в алгоритме мы все изображения приводим к квадратному виду). Пусть

$$D_1^f(y) = \frac{1}{y+1} \cdot \sum_{x=0}^y f(y-x, x), \quad D_2^f(y) = \frac{1}{m-y} \cdot \sum_{x=y}^{m-1} f(m-1+y-x, x), \quad (2)$$

где $y = 0, 1, 2, \dots, m-1$. Функции $D_1^f(y)$ и $D_2^f(y)$ представляют собой средние значения яркостей точек, расположенных по диагоналям.

Для вычисления средних значений яркости точек, расположенных на диагоналях другого направления, достаточно транспонировать матрицу изображения и воспользоваться формулами (2).

С определенной точностью введенные величины $M_y^f(x)$, $M_x^f(y)$, $D_1^f(y)$ и $D_2^f(y)$ аппроксимируют (суммирование ведется только по вертикальным, горизонтальным и диагональным прямым) преобразование Радона изображения f . Поэтому можно попытаться использовать их в задачах классификации изображений. Эти величины дают нам $4n$ признаков изображения, которые достаточно легко вычисляются.

Дадим следующее пояснение к тому, что по формулам (1) и (2) будут вычисляться величины, указывающие на наличие определенного фрагмента рисунка. Для простоты предположим, что входное изображение на черном фоне ($f(x, y) = 0$) содержит отрезок белого цвета ($f(x, y) = 255, x = x_0, y_0 \leq y \leq y_1$). Тогда

$$M_y^f(x_0) = 255 \cdot \frac{y_1 - y_0 + 1}{m},$$

а при $x \neq x_0$ значения $M_y^f(x) = 0$. Это указывает на то, что на линии $x = x_0$ есть точки с высокой яркостью.

С другой стороны, $M_x^f(y) = 255/n$ при $y_0 \leq y \leq y_1$, и $M_x^f(y) = 0$ для остальных y . Следовательно, точки на этой линии с высокой яркостью сосредоточены на отрезке $y_0 \leq y \leq y_1$. Тем самым строение функций $M_y^f(x)$ и $M_x^f(y)$ указывает на наличие отрезка белого цвета.

Как уже было сказано, помимо преобразования Радона, мы используем функции Стиклера. Для случая изображений определим их так. Зададим натуральное число $q = 2k + 1$, $k = 0, 1, 2, \dots$, и для каждой точки (x, y) , $x = k, k+q, \dots, k+[n/q]q-q$, $y = k, k+q, \dots, k+[m/q]q-q$, положим

$$S_q(x, y) = \frac{1}{q^2} \sum_{u=-k}^k \sum_{v=-k}^k f(x+u, y+v).$$

Данная величина определяет осредненное изображение. При этом $S_q(x, y) = f(x, y)$, когда $q = 1$ ($k = 0$). На рис. 1 можно видеть результат такого преобразования. Заметим, что общие очертания изображенного автомобиля легко угадываются, что дает возможность использования полученных признаков в задачах классификации.

Рис. 1. Изображение признаков $S_q(x, y)$ Fig. 1. Image of $S_q(x, y)$ features

3. Описание алгоритма и результатов его работы

Мы будем рассматривать задачу определения наличия на фотоснимке изображения автомобиля (бинарная классификация). Для этого был сформирован обучающий набор изображений автомобилей, снятых с разных ракурсов, в том числе взятый с ресурса <https://www.kaggle.com/>. Помимо этого, на основе различных фотографий городских улиц, зданий, парков и т. д. были сгенерированы фрагменты этих снимков для обучающего набора снимков, не содержащих изображение автомобиля. В итоге мы получили 1776 снимков, содержащих изображение автомобиля, и 1800 снимков, не содержащих изображения автомобиля. Далее для каждого изображения были выполнены следующие вычисления.

- Изображение `img` приводится к размеру 200×200 точек
`res = cv2.resize(img, (200, 200), interpolation=cv2.INTER_CUBIC)`.
- Изображение `res` переводится в оттенки серого цвета
`gray = cv2.cvtColor(res, cv2.COLOR_BGR2GRAY)`.
- Изображение `gray` разбивается на 4 блока размера 100×100 точек: `frame00`, `frame01`, `frame10` и `frame11`. Для каждого из них по формулам (1), (2) вычисляется массив значений, после чего эти данные записываются в текстовый файл `dataset.csv`.

Для построения модели классификации входного изображения мы воспользовались методами машинного обучения, а также их программными реализациями в библиотеке `Scikit-learn` [21]. Для каждого отдельного метода обучение производилось на основе сформированного файла `dataset.csv`. Вся выборка случайным образом делилась в соотношении 7:3 на обучающую часть и тестовую. Результат проверки можно увидеть в табл. 1 (количество признаков 800).

Таблица 1 / Table 1

Результат использования величин (1) и (2)

The result of using values (1) and (2)

Метод	Процент правильного определения класса на тестовой выборке
KNeighborsClassifier	95.34
Naive_bayes.GaussianNB	59.83
DecisionTreeClassifier	88.63
LinearDiscriminantAnalysis	86.30
LinearSVC	86.57
RandomForestClassifier	95.71

Для улучшения результата мы добавили к признакам, вычисляемым по формулам (1) и (2), новые параметры, которые рассчитываются следующим образом. Сначала выполняется фильтрация изображения `gray` с целью выделения границ перепадов уровня яркости



`edges = cv2.Canny(gray, 50, 150, apertureSize = 3).`

На рис. 2 видим результат этих преобразований.

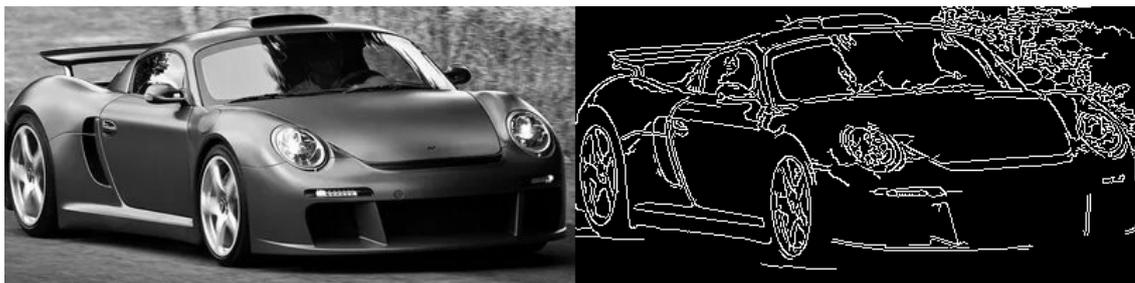


Рис. 2. Выполнение преобразования `cv2.Canny()`
 Fig. 2. Performing the `cv2.Canny()` conversion

После этого изображение разбивается на блоки размером $q \times q$. В каждом таком блоке формируется набор светлых точек в виде массива, содержащего пару $[x_i, y_i]$, где y_i — номер строки и x_i — номер столбца i -й светлой точки. Затем этот набор точек аппроксимируется прямолинейным отрезком, который описывается уравнением $y = a \cdot x + b$. Параметры a, b вычисляются методом наименьших квадратов

$$a = \frac{r \sum_{i=1}^r x_i y_i - \sum_{i=1}^r x_i \cdot \sum_{i=1}^r y_i}{r \sum_{i=1}^r x_i^2 - \left(\sum_{i=1}^r x_i\right)^2}, \quad b = \frac{1}{r} \left(\sum_{i=1}^r y_i - a \cdot \sum_{i=1}^r x_i \right),$$

где r — количество светлых точек в блоке.

Данные величины могут служить признаком на изображении, так как эти отрезки приближают набор белых точек в блоке (рис. 3).

Введение новых параметров несколько улучшило результаты распознавания, при этом количество признаков возросло до 1600 (нами было взято $q = 10$). Помимо доли правильных ответов, нас интересовало и время, которое необходимо той или иной модели для классификации изображения. Поэтому в третьем столбце табл. 2 мы поместили соответствующие значения. Время измерялось на компьютере с процессором AMD® A8-4555m apu with radeon(tm) hd graphics x 4.



Рис. 3. Аппроксимация набора светлых точек прямолинейным отрезком
 Fig. 3. Approximation of a set of light points by a straight line segment

Таблица 2 / Table 2

Результаты использования дополнительных признаков
 Results of using additional features

Метод	Процент правильного определения класса на тестовой выборке	Время выполнения предсказания на одном изображении, с
KNeighborsClassifier	96.64	0.0476
Naive_bayes.GaussianNB	63.00	0.0201
DecisionTreeClassifier	88.25	0.0197
LinearDiscriminantAnalysis	84.43	0.0199
LinearSVC	93.38	0.0200
RandomForestClassifier	95.61	0.0313



Для сравнения нами была сконструирована сверточная нейронная сеть на базе библиотеки Keras со следующей структурой:

Layer (type)	Output Shape	Param #
conv2d_9 (Conv2D)	(None, 198, 198, 18)	180
max_pooling2d_9 (MaxPooling 2D)	(None, 99, 99, 18)	0
conv2d_10 (Conv2D)	(None, 97, 97, 12)	1956
max_pooling2d_10 (MaxPoolin g2D)	(None, 48, 48, 12)	0
conv2d_11 (Conv2D)	(None, 44, 44, 9)	2709
max_pooling2d_11 (MaxPoolin g2D)	(None, 22, 22, 9)	0
flatten_3 (Flatten)	(None, 4356)	0
dense_3 (Dense)	(None, 2)	8714

```
=====  
Total params: 13,559  
Trainable params: 13,559  
Non-trainable params: 0
```

Как видим, в ней общее количество параметров равно 13 599. Обучение сети проходило по тому же набору изображений, что и вычисление признаков и формирование файла `dataset.csv`. В итоге построенная сеть дала 97.69% правильного определения класса изображения на тестовой выборке. Заметим, что мы рассматривали разные варианты строения сети. Вышеприведенная структура сети показала лучший результат. При этом процент правильной классификации изображений из тестовой выборки лежал в диапазоне от 93.7% до 97.69%. Время выполнения предсказания для одного изображения размера 720×1280 точек равнялось 0.0722 с.

Другим подходом к решению различных задач компьютерного зрения является использование предварительно обученной сети. Для нашего случая мы за основу взяли сверточную нейронную сеть, обученную на наборе ImageNet (1.4 млн изображений, разбитых на 1000 разных классов). Мы воспользовались архитектурой VGG16, которая входит в состав фреймворка Keras. Ее можно импортировать из модуля `keras.applications` [22]. Так как набор классов содержит в себе виды различных автомобилей, то мы не стали добавлять плотные слои в нейронную сеть VGG16 и доучивать ее на нашем наборе снимков. Определив выход нейронной сети с помощью метода `predict()` на нашем наборе изображений автомобилей, мы вычислили процент правильно угаданных снимков. Он оказался равным 98.42%. Время выполнения предсказания для одного изображения в среднем составило 0.652 с.

В качестве примера применения разработанного алгоритма рассмотрим задачу нахождения на входном фотоснимке фрагмента, содержащего изображение автомобиля. Для решения данной задачи воспользуемся следующим способом. Пусть снимок имеет размеры W точек в ширину и H точек в высоту. Будем перебирать все фрагменты, имеющие размеры соответственно $[\alpha \cdot W]$ и $[\beta \cdot H]$, где $[x]$ — целая часть числа x , $\alpha < 1$ — некоторое положительное число. Перебор будем осуществлять сверху вниз и слева направо с некоторым шагом q . Для каждого такого фрагмента с помощью построенной модели определяем наличие в нем



изображения автомобиля и обводим этот фрагмент прямоугольником определенного цвета, если автомобиль найден (рис. 4).



Рис. 4. Результат поиска фрагментов с автомобилем
Fig. 4. The result of the search for fragments with a car

Заключение

В работе был описан метод выделения признаков на изображениях, основанный на применении интегральных преобразований. На нескольких иллюстрирующих примерах было показано, что эти признаки вполне подходят для их использования в задачах классификации. Сравнение разработанного подхода со сверточными нейронными сетями позволяет сказать, что он им не сильно уступает по такому показателю, как процент правильных предсказаний. При этом по времени выполнения предсказания представленный в статье метод работает от 2.5 до 140 раз быстрее в зависимости от используемой модели.

Список литературы

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений. Москва : Техносфера, 2005. 1072 с.
2. Борзов С. М., Потатуркин А. О., Потатуркин О. И., Федотов А. М. Исследование эффективности классификации гиперспектральных спутниковых изображений природных и антропогенных территорий // Автометрия. 2016. Т. 52, № 1. С. 3–14. <https://doi.org/10.15372/AUT20160101>, EDN: VSSHQT
3. Дорогов А. Ю., Курбанов Р. Г., Разин В. В. Быстрая классификация JPEG-изображений // Интернет-математика 2005. Автоматическая обработка веб-данных. Москва, 2005. С. 147–172.
4. Осокин В. В., Аипов Т. Д., Ниязова З. А. О классификации изображений и музыкальных файлов // Интеллектуальные системы. Теория и приложения. 2015. Т. 19, № 1. С. 49–70. EDN: UWLELY
5. Купенова Э. М., Кашницкий А. В. Метод случайных лесов в задачах классификации спутниковых снимков // Вестник Тверского государственного университета. Серия: География и геоэкология. 2018. № 3. С. 99–107. EDN: VRYRGY
6. Гончарова Е. Ф., Гайдель А. В. Методы отбора признаков для задач классификации изображений земной поверхности // Информационные технологии и нанотехнологии (ИТНТ-2017) : сб. тр. III международной конференции и молодежной школы. Самара : Новая техника, 2017. С. 535–540. EDN: YQLPDZ
7. Туркова М. А., Гайдель А. В. Корреляционные признаки для классификации текстурных изображений // Информационные технологии и нанотехнологии : сб. тр. IV международной конференции и молодежной школы. Самара : Новая техника, 2018. С. 595–599. EDN: XMWXVZ
8. Чуличков А. И., Морозова И. В. Классификация размытых изображений и оценка параметров системы регистрации методами морфологического анализа // Интеллектуальные системы. 2005. Т. 9, № 1–4. С. 321–344. EDN: LZDWRV
9. Борзов С. М., Потатуркин О. И. Классификация типов растительного покрова по гиперспектральным данным дистанционного зондирования земли // Вестник Новосибирского государственного университета. Серия: Информационные технологии. 2014. Т. 12, вып. 4. С. 13–22. EDN: TMFZOJ



10. Пестунов И. А., Рылов С. А., Мельников П. В. Классификация гиперспектральных изображений высокого пространственного разрешения // Journal of Siberian Federal University. Engineering & Technologies. 2018. Т. 11, вып. 1. С. 69–76. <https://doi.org/10.17516/1999-494X-0010>, EDN: [YPSWQC](#)
11. Китаев Д. Н. Сравнение сверточной нейронной сети и метода главных компонент в задаче классификации текстурных изображений // LXVIII Молодежная научная конференция, посвященная 20-летию со дня начала эксплуатации Международной космической станции : тез. докл. / отв. ред. А. Б. Прокофьев. Самара : Изд-во Самарского ун-та, 2018. С. 88–89. EDN: [FNYNOI](#)
12. Голубков А. М. Бинарная классификация изображений на примере задачи распознавания лиц // Известия СПбГЭТУ «ЛЭТИ». 2018. № 7. С. 26–30. EDN: [VJYUBZ](#)
13. Нежевенко Е. С., Феоктистов А. С., Дашевский О. Ю. Нейросетевая классификация гиперспектральных изображений на основе преобразования Гильберта – Хуанга // Автометрия. 2017. Т. 53, № 2. С. 79–84. <https://doi.org/10.15372/AUT20170209>, EDN: [YKFYYR](#)
14. Васкан В. Д. Обзор архитектур сверточных нейронных сетей для задачи классификации изображений // ИТ-Стандарт. 2021. № 3 (28). С. 34–39. EDN: [WQQUFN](#)
15. Солдатова О. П., Гаршин А. А. Применение сверточной нейронной сети для распознавания рукописных цифр // Компьютерная оптика. 2010. Т. 34, № 2. С. 252–260. EDN: [MOTHJX](#)
16. Ле Мань Ха. Сверточная нейронная сеть для решения задачи классификации // Труды Московского физико-технического института (НИУ). 2016. Т. 8, № 3. С. 91–97. EDN: [WWMDBT](#)
17. Азаров И. С., Прокопеня А. С. Сверточные нейронные сети для распознавания изображений // Big Data and Advanced Analytics. 2020. № 6–1. С. 271–280. EDN: [HBCXHL](#)
18. Сикорский О. С. Обзор сверточных нейронных сетей для задачи классификации изображений // Новые информационные технологии в автоматизированных системах. 2017. № 20. С. 37–42. EDN: [YNADUJ](#)
19. Deans S. R., Roderick S. The Radon Transform and Some of its Applications. New York : John Wiley & Sons, 1983. 289 p.
20. Жук В. В., Кузютин В. Ф. Аппроксимация функций и численное интегрирование. Санкт-Петербург : Изд-во Санкт-Петербургского ун-та, 1995. 352 с.
21. Воронина В. В., Михеев А. В., Ярушкина Н. Г., Святков К. В. Теория и практика машинного обучения : учеб. пособие. Ульяновск : УлГТУ, 2017. 290 с. EDN: [QNQXJP](#)
22. Шолле Ф. Глубокое обучение на Python. Санкт-Петербург : Питер, 2018. 400 с. (Библиотека программиста).

References

1. Gonzalez R., Woods R. Digital Image Processing. Pearson/Prentice Hall, 2008. 954 p. (Russ. ed.: Moscow, Tekhnosfera, 2005. 1072 p.).
2. Borzov S. M., Potaturkin A. O., Potaturkin O. I., Fedotov A. M. Analysis of the efficiency of classification of hyperspectral satellite images of natural and man-made areas. *Optoelectronics, Instrumentation and Data Processing*, 2016, vol. 52, iss. 1, pp. 1–10. <https://doi.org/10.3103/S8756699016010015>, EDN: [WWFRIF](#)
3. Dorogov A. Yu., Kurbanov R. G., Razin V. V. Quick classification of JPEG images. *Internet Mathematics 2005. Automatic Processing of Web Data*. Moscow, 2005, pp. 147–172 (in Russian).
4. Osokin V. V., Aipov T. D., Niyazova Z. A. About the classification of images and music files. *Intelligent Systems. Theory and Applications*, 2015, vol. 19, iss. 1, pp. 49–70 (in Russian). EDN: [UWLELY](#)
5. Kупенова Е. М., Кашнитский А. В. The method of random forests in the classification tasks of satellite images. *Herald of Tver State University. Series: Geography and Geoecology*, 2018, iss. 3, pp. 99–107 (in Russian). EDN: [VRYRGY](#)
6. Goncharova E. F., Gaidel A. V. Methods of feature selection for problems of classification of images of the Earth's surface. *Informatsionnye tekhnologii i nanotekhnologii (ITBT-2017)* [Information Technologies and Nanotechnology (ITNT-2017): Proceedings of the III International Conference and Youth school]. Samara, Novaya Tekhnika, 2017, pp. 535–540 (in Russian). EDN: [YQLPDZ](#)
7. Turkova M. A., Gaidel A. V. Correlation features for classification of texture images. *Informatsionnye tekhnologii i nanotekhnologii* [Information Technologies and Nanotechnology: Proceedings of the IV International Conference and Youth School]. Samara, Novaya Tekhnika, 2018, pp. 595–599 (in Russian). EDN: [XMWXVZ](#)
8. Chulichkov A. I., Morozova I. V. Classification of blurred images and estimation of registration system parameters by morphological analysis methods. *Intelligent Systems*, 2005, vol. 9, iss. 1–4, pp. 321–344 (in Russian). EDN: [LZDWRV](#)



9. Borzov S. M., Potaturkin O. I. Vegetative cover type classification using hyperspectral remote sensing. *Vestnik NSU. Series: Information Technology*, 2014, vol. 12, iss. 4, pp. 13–22 (in Russian). EDN: [TMFZOJ](#)
10. Pestunov I. A., Rylov S. A., Melnikov P. V. Classification of hyperspectral images with high spatial resolution. *Journal of Siberian Federal University. Engineering & Technologies*, 2018, vol. 11, iss. 1, pp. 69–76 (in Russian). <https://doi.org/10.17516/1999-494X-0010>, EDN: [YPSWQC](#)
11. Kitaev D. N. Comparison of a convolutional neural network and the principal component method in the problem of classification of texture images. In: Prokofiev A. B. (ed.) *Abstracts of LXVIII Youth Scientific Conference Dedicated to the 20th Anniversary of the Start of Operation of the International Space Station*. Samara, Samara State University Publ., 2018, pp. 88–89 (in Russian). EDN: [FNYNOI](#)
12. Golubkov A. M. Binary classification of images using the example of the face recognition problem. *Izvestiya SPbGETU "LETI"*, 2018, iss. 7, pp. 26–30 (in Russian). EDN: [VJYUBZ](#)
13. Nezhevenko E. S., Feoktistov A. S., Dashevsky O. Yu. Neural network classification of hyperspectral images of the basis of the Hilbert – Huang transform. *Optoelectronics, Instrumentation and Data Processing*, 2017, vol. 53, iss. 2, pp. 165–170. <https://doi.org/10.3103/S8756699017020091>, EDN: [XNIJBJ](#)
14. Vaskan V. D. Overview of convolutional neural network architectures for the image classification problem. *IT Standard*, 2021, iss. 3 (28), pp. 34–39 (in Russian). EDN: [WQQUFN](#)
15. Soldatova O. P., Garshin A. A. The use of a convolutional neural network for recognizing handwritten numbers. *Computer Optics*, 2010, vol. 34, iss. 2, pp. 252–260 (in Russian). EDN: [MOTHJX](#)
16. Le Man Ha. Convolutional neural network for solving the classification problem. *Proceedings of Moscow Institute of Physics and Technology*, 2016, vol. 8, iss. 3, pp. 91–97 (in Russian). EDN: [WWMDBT](#)
17. Azarov I. S., Prokopenya A. S. Overview of convolutional neural networks for image recognition. *Big Data and Advanced Analytics*, 2020, iss. 6–1, pp. 271–280 (in Russian). EDN: [HBCHXL](#)
18. Sikorsky O. S. Overview of convolutional neural networks for the problem of image classification. *New Information Technologies in Automated Systems*, 2017, iss. 20, pp. 37–42 (in Russian). EDN: [YNADUJ](#)
19. Deans S. R., Roderick S. *The Radon Transform and Some of its Applications*. New York, John Wiley & Sons, 1983. 289 p.
20. Zhuk V. V., Kuzyutin V. F. *Approksimatsiya funktsiy i chislennoye integrirvaniye* [Approximation of Functions and Numerical Integration]. St. Petersburg, St. Petersburg University Publ., 1995. 352 p. (in Russian).
21. Voronina V. V., Mikheev A. V., Yarushkina N. G., Svyatov K. V. *Teoriya i praktika mashinnogo obucheniya* [Theory and Practice of Machine Learning]. Ulyanovsk, Ulyanovsk State Technical University Publ., 2017. 290 p. (in Russian). EDN: [QNQXJP](#)
22. Chollet F. *Deep Learning with Python*. New York, Manning Publications Co., 2018. 374 p. (Russ. ed.: St. Petersburg, Piter, 2018. 400 p.).

Поступила в редакцию / Received 27.02.2023

Принята к публикации / Accepted 08.04.2023

Опубликована / Published 30.08.2024