DOI: 10.15514/ISPRAS-2023-35(6)-9

Классификация текста растрового документа по признаку начертания

[1,2] *Д.Е. Копылов,* ORCID: 0009-0000-6348-4004 <*it-daniil@yandex.ru*>
[1,2] *А.А. Михайлов,* ORCID: 0000-0003-4057-4511 <*mikhailov@icc.ru*>

[1] *Институт динамики систем и теории управления СО РАН,*
664033, Россия, г. Иркутск, ул. Лермонтова, д. 134,
[2] *Институт системного программирования им. В.П. Иванникова РАН,*
109004, Россия, г. Москва, ул. А. Солженицына, д. 25.

Аннотация. При выделении логической структуры документов используются ряд свойств, одним из которых является полужирное начертание слов текста. Полужирным начертанием в документах часто выделяют заголовки, определяемые слова, названия колонок в таблицах. В данной работе предложен метод классификации текста по жирности начертания, который состоит из последовательности шагов. На первом шаге проводится бинаризация всего изображения. Целью данного шага является разделение пикселей изображения на пиксели текста и фона. Вторым шагом проводится оценка каждого слова. В качестве результата возвращается величина, характеризующая толщину основного штриха символа в данном слове. На последнем шаге проводится кластеризация оценок на два кластера: жирный текст и обычный. Предложенный метод был реализован и протестирован на трех наборах данных, исходный код опубликован в открытом репозитории.

Ключевые слова: анализ документов; растровые документы; классификация текста.

Для цитирования: Копылов Д.Е., Михайлов А.А. Классификация текста растрового документа по признаку начертания. Труды ИСП РАН, том 35, вып. 6, 2023 г., стр. 157–166. DOI: 10.15514/ISPRAS–2023–35(6)–9.

Classification of Printed Text on Raster Documents

[1,2] *D.E. Kopylov,* ORCID: 0009-6348-0000-4004 <*it-daniil@yandex.ru*>
[1,2] *A.A. Mikhailov,* ORCID: 0000-0003-4057-4511 <*mikhailov@icc.ru*>

[1] *Matrosov Institute for System Dynamics and Control Theory*
of the Siberian Branch of the Russian Academy of Sciences,
134, Lermontova st., Irkutsk, 664033, Russia.
[2] *Ivannikov Institute for System Programming of the Russian Academy of Sciences,*
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.

Abstract. When highlighting the logical structure of documents, a number of properties are used, one of which is the bold style of text words. In documents, headings, defined words, and column names in tables are often highlighted in bold. This paper proposes a method for classifying text by boldness, which consists of a sequence of steps. The first step is binarization of the entire image. The purpose of this step is to separate the image pixels into text and background pixels. The second step is to evaluate each word. The result is returned a value characterizing the thickness of the main stroke of the character in the given word. At the last step, the ratings are clustered into two clusters: bold text and regular. The proposed method was implemented and tested on three data sets, and the source code was published in an open repository.

Keywords: document analysis, raster documents, text classification.

For citation: Kopylov D.E., Mikhailov A.A. Classification of printed text on raster documents. *Trudy ISP RAN/Proc. ISP RAS*, vol. 35, issue 6, 2023. pp. 157-166 (in Russian). DOI: 10.15514/ISPRAS-2023-35(6)-9

1. Введение

Человеческая деятельность неразрывно связана с обработкой документов. Люди ежедневно имеют дело с товарными чеками, инструкциями, брошюрами и т.п. С гораздо большим объемом информации работают предприятия и организации. Процессы, протекающие в них, осуществляются за счет документооборота. К таким документам можно отнести приказы, кадровые документы, счета фактур и т.д.

С развитием компьютеров работать с документами стало гораздо проще. Сейчас компьютер способен обрабатывать автоматически некоторые типы документов, но большую часть до сих пор нельзя обработать без участия человека. Научить компьютер «понимать документы» (document understanding) – это задача, которую пытаются решить многие исследователи. Усилия исследователей направлены на разные стороны этой проблемы, важной из которых является восстановление логической структуры документа.

Для построения логической структуры документа используется различные свойства текста, одним из таких свойств является полужирное начертание (в статье «жирное начертание» будет использоваться как синоним к полужирному начертанию). Жирным начертанием выделяются заголовки или определяемые слова. Часто это одно из самых главных свойств, а порой единственное, по которым можно восстановить структуру документа. При обработке документов в форматах DOC, HTML, XML и т.п. получить информацию о жирности не составляет труда. Это не относится к документам в растровом формате, в которых информация не представлена в явном виде. Говоря о жирном начертании слов, заданных в виде изображения, нужно понимать, что речь идет о толщине символов (конкретно о толщине основных штрихов, которые вносят основной вклад в восприятия символа). Насыщенность символа определяется по основным штрихам и несет полезную информацию, только если есть слова менее насыщенные. Так, например, если текст целиком написан жирным шрифтом, то в таком случае он является основным. На восстановление логической структуры в этом случае полужирное начертание никак не повлияет.

В схожих работах, как правило, классифицируются шрифты в целом. Для этого обычно используются подходы на основе нейронных сетей. Так делают авторы работ, например, [1]–[2]. Для распознавания шрифтов хорошие результаты показывают нейронные сети с архитектурой трансформер.

Цель нашего исследования является классификация текста исключительно по признаку жирного начертания. Использовать упомянутые нейронные сети и из полученной информации брать только начертание будет избыточно сложным. Обучение нейронной сети исключительно для распознавания жирного текста на изображение может осуществляться только для документов, шрифты которых заранее известны (причем жирное начертание одного не похоже на обычное начертание другого).

В связи с этим, для решения задачи выбран подход с использованием математических моделей и статистик. Таким же образом поступают авторы работы [3]. Авторы предлагают разные алгоритмы для определения жирного шрифта, курсива, высоты символа. В работе приводятся несколько примеров работы разработанных алгоритмов, но отсутствует этап тестирования. Помимо этой работы, нам не удалось найти исследования, содержащие алгоритмы и их тестирование.

2. Предлагаемая схема работы метода

Нами была предложена схема работы подхода представленная на рис. 1. Предложенный метод был реализован на языке Python и опубликован в открытом репозитории исходных кодов[1]. Далее кратко будет передана идея работы метода, а в подпунктах 2.1 – 2.3 подробно будут описаны ключевые шаги метода.

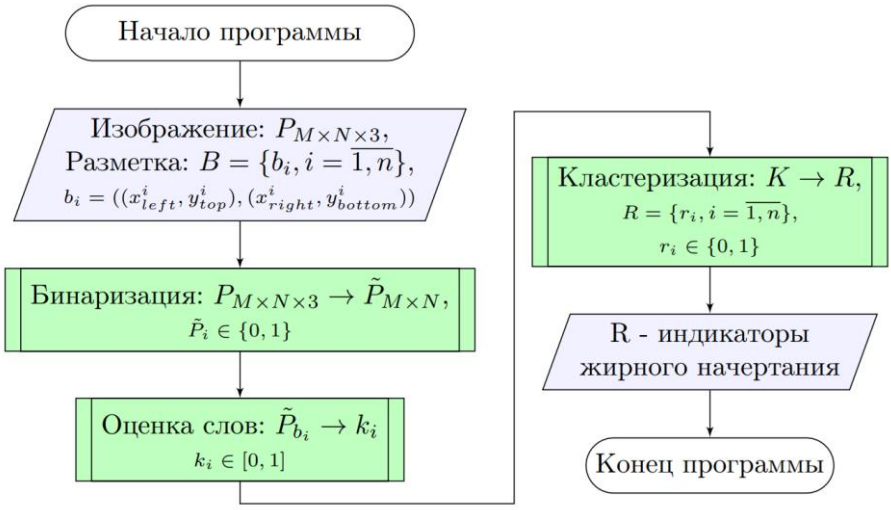

Рис. 1. Схема работы метода
Fig. 1. Scheme of method

На вход подается цветное изображение и координаты слов. На выходе алгоритм возвращает набор индикаторов, характеризующих жирный шрифт это или нет.

На первом шаге работы проводится бинаризация всего изображения. Целью данного шага является разделение пикселей изображения на пиксели текста и фона.

Вторым шагом проводится оценка каждого слова. В качестве результата возвращается величина, характеризующая толщину основного штриха символа в данном слове.

На последнем шаге проводится кластеризация оценок на два кластера: жирный текст и обычный.

2.1 Бинаризация

Для определения размеров требуется знать границу объекта, это относится и к толщине штриха. В процессе оценки так или иначе будет возникать вопрос о принадлежности пикселя к фону или символу. Для этого используется бинаризация, после которой становится ясно, где находятся границы символа. Неправильное выставление порога для бинаризации может привести к ухудшению изображения. Так, например, при высоком пороге бинаризации велик шанс сделать обычный текст неотличимым от жирного. Также стоит помнить, что фон на изображениях не обязательно будет абсолютно белым, а текст абсолютно черным. Исходя из вышесказанного, стоит отдавать предпочтение адаптивной бинаризации.

При работе с текстом распространена бинаризация Otsu [4], которая минимизирует взвешенную дисперсию двух классов (оттенков пикселей текста и пикселей фона). В нашей схеме используется адаптивная бинаризация «выделение впадины» [5]. В отличие от бинаризации Otsu, она берет в учет «впадину», возникающую в спектре изображения. В бимодальных распределениях один из пиков может иметь вес в несколько раз больше, чем

[1] https://github.com/Dann38/bold_classifier

другой. В таком случае бинаризация Otsu выставит порог, ближе к пику с наибольшим весом. Такой эффект возникает, например, при идеальном белом фоне и размытом тексте. Обрабатывая такой текст с помощью бинаризации Otsu, получаем весь текст похожий на жирное начертание. В случае бинаризации методом «выделения впадины» порог сдвигается в сторону «впадины» на графике спектра изображения, что избавляет от этого неприятного эффекта. Для бинаризации «выделение впадины» формула расчета порога T выглядит следующим образом:

$$T = argmax_{t\in(0,255)}\omega_3[\omega_1(t)\mu_1^2(t) + \omega_2(t)\mu_2^2(t)],$$

где $\omega_1(t), \omega_2(t)$ – вес темных и светлых пикселей, $\mu_1(t), \mu_2(t)$ – математическое ожидание темных и светлых пикселей, $\omega_3(t)$ – вес впадины. Для впадины дополнительно необходимо указать радиус. В предлагаемой схеме используется радиус равный 5. На данных, которые приведены в разделе тестирование, такой радиус показал себя лучше всего. Вопрос выбора бинаризации и ее параметров достоин отдельного исследования.

2.2 Оценка жирности начертания слова

Для оценки жирности начертания в данной работе используются статистические величины и их комбинации. Негативное влияния на статистические оценки оказывает шум и выбросы. К таким шумам можно отнести белое пространство вне области строчных букв – это область между верхними и нижними выносными элементами символов. С точки зрения математики, бинаризованное изображение — это матрица, состоящая из 0 и 1. Необходимо найти номер строки матрицы, где заканчиваются верхние выносные элементы, а также номер, где начинаются нижние выносные элементы. Сделать это можно, используя следующие формулы:

$$i_{top} = argmax_{i=\overline{1,m}}(p_{i-1} - p_i), i_{bottom} = argmax_{i=\overline{1,m}}(p_{i+1} - p_i),$$

$$p_i = mean_i(P),$$

где P – матрица бинаризованного изображения слова с числом строк равным m , $mean_i(\cdot)$– среднее значение i-й строки матрицы.

Недостатком данных подходов могут служить акронимы, цифры и текст, написанный в верхнем регистре. При этом стоит отметить, что маловероятно, что акроним из 2–5 букв будет являться, например, отдельно стоящим заголовком.

Другим примером шума служат пробелы между буквами. Для их удаления, рассчитывается среднее значение для каждого столбца матрицы изображения, а затем удаляются те, где среднее близко к единице. Под близостью к единице понимается значение 0.95. Для символов “т”, “г” значение порога критично, так как верхняя часть этих символов может быть достаточно мала, и будет признана пробелом. Среднее значение в этой области находится в диапазоне 0.90-0.95. По этой причине и выбирается значение 0.95.

Говоря непосредственно о способах оценки и статистических характеристиках с ними связанных, можно выделить три подхода:

1) Среднее значение интенсивности пикселей;
2) Медианное значение интенсивности пикселей;
3) Отношение периметра к площади символов.

Достаточно простым является первый вариант. Чем ниже оценка, тем более насыщенное слово. Чем более насыщенное слово, там вероятнее оно имеет жирное начертание. Недостатком такого подхода является то, что некоторые символы являются более насыщенными, чем другие.

Более устойчивой к таким выбросам является медиана, лежащая в основе второго подхода.

В третьем подходе используется идея, что оценить насыщенность слова можно как отношение двух характеристик, а именно:

- периметр символов (контур),
- площади символов.

В третьем подходе насыщенность измеряется как среднее значение, но при этом в учет берется форма символов. Учет формы обусловлен тем, что для прорисовки букв используется разное число пикселей. Так, например, для буквы «г» нужно меньше темных пикселей, чем для буквы «в». Оценка находится по следующей формуле:

$$w = \frac{c}{s},$$

где c – характеристика контура, s – характеристика площади.

Для удобства и эффективности вычислений эти оценки заменяются их эквивалентами. Для оценки площади используется оценка:

$$s = 1 - mean(P), P = \{p_{ij}\}$$

а для периметра оценка:

$$c = mean(dP), dP = \{p_{i+1,j} - p_{ij}\},$$

где P – матрица изображения слова, p_{ij} – элемент, стоящий на пересечении i-й строки и j-го столбца, dP – аналог изображения контура, $mean(\cdot)$– среднее значение.

При изменении начертания слова, периметр у букв изменяется непропорционально площади. В таком случае у полужирного начертания по сравнению с обычным значение w будет ниже. При прочих равных условиях данный подход показывает себя лучше, чем оценка среднего значения или поиск медианы в выборке пикселей изображений (рис. 2).

2.3 Кластеризация

После получения оценок последним шагом метода является разбиение слов на кластеры. Основной проблемой при проведении кластеризации является определения числа кластеров. В работе делается допущение, что на странице присутствует только два типа начертания. Свойство курсивного начертания не учитывается. Наличие промежуточной насыщенности также исключается, так как является неоправданно сложным. Начертание формул и рукописного текста на документе считается неизвестным и не учитывается при оценке подходов. Таким образом, количество кластеров не может превышать двух. Возможность наличия всего одного кластера обусловлена тем, что весь текст может быть написан обычным начертанием. Для определения числа кластеров проверяется гипотеза об однородности выборки. В данной работе используется критерий Duda и Hart [6].

Для оценки определяется значение

$$F_{\frac{2}{1}} = \frac{w_2}{w_1},$$

где w_1 – сумма квадратов внутрикластерного расстояния между двумя классами, w_2– сумма квадратов отклонения от среднего значения всех оценок. Это значение сравнивается с:

$$F_{\text{кр}} = 1 - \frac{2}{\pi \cdot p} - z_{(1-\alpha)} \cdot \sqrt{2 \cdot \frac{1 - \frac{8}{\pi^2 \cdot p}}{n \cdot p}}$$

где p – размерность кластеризуемого пространства, $z_{1-\alpha}$ – квантиль нормального распределения, n – число кластеризуемых объектов. При $F_{\text{кр}} < F_{\frac{2}{1}}$ отвергается гипотеза о

присутствии двух кластеров, а значит, нет оснований утверждать, что на изображении присутствует жирный шрифт.

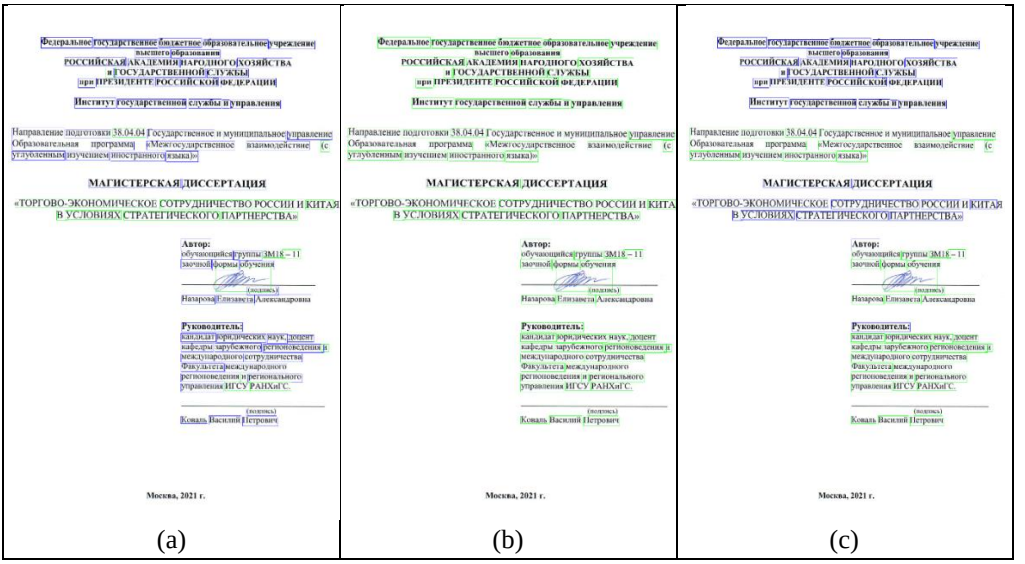

Рис. 2. Результат работы классификатора (зеленым – обычный шрифт, синим – жирный) с использованием в качестве оценки
(a) – среднего значения, (b) – медианного значения, (c) – отношение периметра к площади
Fig. 2. The result of the classifier (green – regular, blue – bold) using as an estimate
(a) – the average value, (b) – the median value, (c) – the ratio of the perimeter to the area

В качестве параметра для описанного выше критерия нужно задать уровень значимости α (вероятность того, что будет отвергнута гипотеза, хотя она верна). В работе выбирается $\alpha = 0.15$. В рамках данной работы параметр подробно не изучался, но при таком значении α чаще будет не найден жирный текст, чем обычный ошибочно выделен жирным.

В работе рассматриваются три метода кластеризации:

- Кластеризация k-средних;
- Спектральная кластеризация;
- Агломеративная кластеризация.

Первые две кластеризации являются достаточно распространёнными [7]. Третий вид кластеризации менее известен. Агломеративная кластеризация выигрывает по сравнению со спектральной и кластеризацией k-средних за счет того, что изначально каждая оценка слова — это отдельный кластер. На каждом следующем шаге объединяются те кластеры, которые ближе друг к другу. На последнем шаге остаются два кластера, которые имеют расстояние между друг другом больше, чем между любым элементом из кластера и его ближайшим соседом. Учитывая, что слова состоят из символов с разной насыщенностью, начинать объединять их лучше по близости оценок.

Кластеризация проводится над векторами, что позволяет помимо оценки текущего слова добавить информацию об оценках предыдущего и следующего слова. Шанс того, что слово окажется написано жирным начертанием выше, если вокруг него слова написаны жирным начертанием. Каждому слову ставится в соответствие вектор:

$$x_i = (k_i, \frac{k_{i-1}+k_i+k_{i+1}}{3}), k_0 = k_1, k_{n+1} = k_n, i = 1, n,$$

где k_i – оценка i-слова, k_{i-1} – оценка предшествующего слова, k_{i+1} – оценка следующего слова.

3. Тестирование

Для тестирования использовались три набора данных, размеченных вручную.

- Первый набор называется «ВКР», он состоит из 30 изображений в следующей пропорции: 5 изображений со сложным текстом (титульные листы и листы содержания), 20 изображений с обычным текстом на которых присутствует текст с жирным начертанием, 5 изображений, не содержащие жирный шрифт. Общее число слов 5420. Число слов жирным начертанием 410 (7.6% от всего текста)
- Второй набор называется «ГОСТ». Он состоит из 30 изображений, полученных при конвертировании PDF ГОСТа в изображение. Особенностью данного набора является то, что изображение имеет более низкое разрешение по сравнению с «ВКР», также кегель текста более мелкий. Общее количество слов 10769. Число слов жирным начертанием 685 (6.3% от всего текста).
- Третий набор данных называется «СКАН». Он состоит из 20 изображений, полученных как сканы одного из учебников. Особенностью таких данных является то, что в них фон не абсолютно белый и текст не абсолютно черный. Также присутствуют небольшие искажение слов. Общее количество слов 5055. Число слов жирным начертанием 407 (7.8% от всего текста).

При тестировании использовалась F_1-мера, являющаяся классической оценкой в анализе данных. Расчет проводился для жирного начертания. Дополнительно был рассчитан показатель точности $Accuracy$ для того, чтобы показать правильность определения значений в целом.

Как было описано ранее, метод оценки слов по жирности начертания состоит из подпрограмм. Ключевым шагом является оценка слов. В табл. 1 приведены сравнения трех вариантов оценок в зависимости от разных данных и кластеризации. Другой важный шаг, это кластеризация слов по оценкам. Сравнение разных методов кластеризаций приведено в табл. 2.

По табл. 2 можно сделать вывод, что по сравнению с представленными подходами, оценка жирности как отношение периметра к площади показал себя лучше в независимости от вида данных, и метода кластеризации.

В табл. 3 приведены значения Accuracy при оценке отношения периметра к площади. Из табл. 3 видно, что нельзя сказать, что один из методов кластеризации превосходит остальные. Для ВКР показала себя лучше агломеративная кластеризация. Эти документы были представлены в хорошем качестве. Для наборов данных СКАН и ГОСТ все не так однозначно.

Если сравнивать только спектральную кластеризацию и кластеризацию k-средних, то стоит отдать предпочтение кластеризации k-средних. Выбор между агломеративной кластеризацией и кластеризацией k-средних будет зависеть от типа документов.

Табл. 2. Оценки f1 и accuracy для трех видов оценок в зависимости от набора данных и метода кластеризации
Table 2. F1 and accuracy estimates for three types of estimates depending on the data set and clustering method

Датасет	Кластеризация	Отношение периметра к площади		Среднее значение		Медианное значение	
		f1	accuracy	f1	accuracy	f1	accuracy
ВКР	Агломеративная	**0.72**	**0.95**	0.07	0.57	0.04	0.66
	k-средних	**0.67**	**0.93**	0.10	0.46	0.08	0.60
ГОСТ	Агломеративная	**0.09**	**0.83**	0.11	0.28	0.11	0.35
	k-средних	**0.20**	**0.84**	0.11	0.28	0.11	0.30

Табл. 3. Оценка показателя точности "accuracy" для трех методов кластеризации в зависимости от набора данных при использование оценку по отношению площади к периметру
Table 3. Accuracy estimation for three clustering methods depending on the data set when using an estimate with respect to area to perimeter

Наборы данных	Агломеративная кластеризация	Спектральная кластеризация	Кластеризация k-средних
ВКР	**0.95**	0.86	0.93
СКАН	0.86	0.83	**0.91**
ГОСТ	0.83	**0.84**	**0.84**

4. Заключение

В данной работе был предложен метод решения задачи классификации строк по признаку жирности начертания. Процесс классификации разбивается на этапы: предобработка изображения, оценка изображений слов и кластеризация слов по полученным оценкам.

В процессе предобработки изображения документа важным оказывается избавление от различных шумов и сохранение насыщенности символов. При оценке насыщенности слова необходимо помнить, что символ состоит из штрихов. Не все штрихи при изменении начертания могут меняться. В случае оценки штрихов необходимо делать акцент на основные из них. Способ оценки с учетом формы символов из всех предложенных в работе является наилучшим. По накопившемся оценкам слов, кластеризация слов может осуществляться с использованием разных методов. Для документов в хорошем качестве рекомендуется агломеративная кластеризация. В случае плохого качества стоит отдать предпочтение кластеризации k-средних.

Используя представленный подход, можно находить в тексте слова, написанные полужирным начертанием. Обладая этой информацией, упрощается процесс восстановления логической структуры документа. Таким способом можно находить заголовки, структуры ключ-значение, заголовки столбцов в таблицах и другие компоненты.

Список литературы / References

[1]. Sandy I.C., Voinea D., Popa A.I. CONTENT: Context Sensitive Transformer for Bold Words Classification. arXiv:2205.07683.

[2]. Bychkov O., Merkulova K., Dimitrov G., Zhabska Y., Kostadinova I., Petrova P., Petrov P., Getova I., Panayotova G. Using Neural Networks Application for the Font Recognition Task Solution. In Proc. of 55th International Scientific Conference on ICEST, 2020. pp. 167-170. doi: 10.1109/ICEST49890.2020.9232788.

[3]. Ladareanu L., Chiroiu V., Bratu, P., Magheti, I. Automatic Text Clustering and Classification Based on Font Geometrical Characteristics. In Proc. of 9th WSEAS International Conference on Automation and Information, 2008, pp. 468-473.

[4]. Otsu N. A threshold selection method from gray-level histograms // IEEE Trans. Sys., Man., Cyber. : journal. — 1979. — Vol. 9. — P. 62—66.

[5]. Xing J., Yang P., Qingge L. Automatic thresholding using a modified valley emphasis. IET Image Processing, vol. 14(3), 2020, pp. 536-544. doi: 10.1049/iet-ipr.2019.0176

[6]. Яцкив И., Гусарова Л. Методы определения количества кластеров при классификации без обучения. The Journal of Transport and Telecommunication Institute, vol. 4(1), 2003. pp. 23-28.

[7]. Бурков А. Машинное обучение без лишних слов. Санкт-Петербург, Питер, 2020, 192 с.

Информация об авторах / Information about authors

Даниил Евгеньевич КОПЫЛОВ – магистрант направления подготовки «Прикладная математика и информатика» Иркутского государственного университета, сотрудник Институт системного программирования им. В.П. Иванникова Российской академии наук, сотрудник Института динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук. Сфера научных интересов: прикладная математика, анализ данных.

Daniil Evgenievich KOPYLOV is master’s student of Irkutsk State University, employee of Ivannikov Institute for System Programming of the Russian Academy of Sciences, employee of Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences. Research interests: applied mathematics, data analysis.

Андрей Анатольевич МИХАЙЛОВ является старшим научным сотрудником лаборатории Комплексных информационных систем Института динамики систем и теории управления имени В.М. Матросова. Его научные интересы включают анализ электронных документов, распознавание образов.

Andrey Anatolievitch MIKHAYLOV is a senior researcher of the Laboratory of information systems of Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences. His research interests include document analysis, image recognition.