НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
март–апрель 2018 Том 18 № 2 ISSN 2226-1494 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
March–April 2018 Vol. 18 No 2 ISSN 2226-1494 http://ntv.ifmo.ru/en

УДК 004.932.2

ИСПОЛЬЗОВАНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ОПРЕДЕЛЕНИЯ НАРУШЕНИЙ ЦЕЛОСТНОСТИ JPEG-ИЗОБРАЖЕНИЙ

А.И. Серова[a], А.И. Спивак[a]

[a] Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация
Адрес для переписки: Aliceinwobderland25@gmail.com

Информация о статье
Поступила в редакцию 20.12.17, принята к печати 16.02.18
doi: 10.17586/2226-1494-2018-18-2-299-306
Язык статьи – русский

Ссылка для цитирования: Серова А.И., Спивак А.И. Использование методов машинного обучения для определения нарушений целостности JPEG-изображений // Научно-технический вестник информационных технологий, механики и оптики. 2018. Т. 18. № 2. С. 299–306. doi: 10.17586/2226-1494-2018-18-2-299-306

Аннотация

Предмет исследования. Проведено исследование нарушений целостности изображений и существующих методов их определения. Предложен метод, позволяющий определять модифицированное изображение, а также источник его модификации. Метод позволяет определять оригинальное изображение и модель камеры, на которую оно было снято. **Метод.** В предлагаемом методе использованы инструменты машинного обучения. Исследованы следующие методы машинного обучения: наивный байесовский классификатор, дерево решений, логистическая регрессия, *k*-ближайших соседей, SVC, random forest. База для обучения модели была образована оригинальными изображениями с веб-сайта www.steves-digicams.com, модифицированными с помощью различных графических редакторов. Предложенный метод использует структуру JPEG-изображения в байтовом представлении, а именно маркеры. В качестве признаков для классификации выступали наличие маркеров и их количество. **Основные результаты.** Обученная модель показала высокий результат классификации – более 95%. Среди исследованных алгоритмов два показали наилучшие результаты – дерево решений и random forest, по критерию стабильности было выбрано дерево решений. **Практическая значимость.** Полученный результат может быть применен на практике в таких областях, как криминалистика и информационная безопасность.

Ключевые слова
машинное обучение, изображения, целостность, информационная безопасность, JPEG

APPLICATION OF MACHINE LEARNING METHODS FOR DETECTING OF JPEG IMAGE INTEGRITY VIOLATIONS

A.I. Serova[a], A.I. Spivak[a]

[a] ITMO University, Saint Petersburg, 197101, Russian Federation
Corresponding author: Aliceinwobderland25@gmail.com

Article info
Received 20.12.17, accepted 16.02.18
doi: 10.17586/2226-1494-2018-18-2-299-306
Article in Russian
For citation: Serova A.I., Spivak A.I. Application of machine learning methods for detecting of JPEG image integrity violations. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2018, vol. 18, no. 2, pp. 299–306 (in Russian). doi: 10.17586/2226-1494-2018-18-2-299-306

Abstract

Subject of Research. The paper presents the study on the JPEG image integrity violations and existing methods of their detection. We propose a method for detection of modified image and the source of its modification. The method gives the possibility to determine the original image and camera model that recorded it. **Method.** The method was developed with the use of machine learning tools. The following machine learning methods have been studied: naive Bayesian classifier, decision tree, logistic regression, *k*-nearest neighbors, SVC, random forest. The base for model training was formed by the original photos from website www.steves-digicams.com that were modified by different graphic editors. The proposed method uses JPEG-image structure in byte view, namely, markers. Availability of markers and their number were suggested as classification features. **Main Results.** The trained model has demonstrated high classification result equal to more than 95%. Among all evaluated algorithms the two ones have shown the best results: decision tree and random forest. Decision tree was chosen as the best one upon stability criterion. **Practical Relevance.** The received result can be practically applicable in the area of forensics and information security.

Keywords
machine learning, images, integrity, information security, JPEG

Введение

Исследование изображений в ходе экспертиз для подтверждения факта отсутствия модификаций техническими средствами обычно основано на визуальном осмотре. В силу развития технологий работы графических редакторов в области качественного редактирования изображений достоверность такого метода недостаточна. Одним из способов повышения достоверности детектирования модификаций изображения является внедрение интеллектуальных методов анализа. Широкое применение изображений в различных отраслях экономики, начиная от страхования и заканчивая правоохранительными органами, обусловливает актуальность разработки метода определения нарушений целостности изображений в области информационной безопасности [1].

Алгоритм JPEG (Joint Photographic Experts Group) в наибольшей степени пригоден для сжатия фотографий и картин, содержащих реалистичные сцены с плавными переходами яркости и цвета [2]. Наибольшее распространение JPEG получил в цифровой фотографии, для хранения и передачи изображений с использованием сети Интернет. Это обусловило выбор именно этого формата для разработки метода.

Целью данной работы является разработка метода определения нарушений целостности JPEG-изображения и выявление его источника (фотокамера или графический редактор) при помощи методов машинного обучения.

Проблема подлинности изображений появилась вместе с изобретением фотоаппаратов [3]. Но если вначале можно было определять оригинальность изображений визуально, то с развитием информационных технологий сложность детектирования модификаций значительно возросла.

С появлением цифровых фотоаппаратов (1981 г.) и разработкой формата JPEG (1991 г.) стали появляться различные методы определения подлинности изображений. Один из самых распространенных существующих решений данной задачи является проверка метаданных – EXIF (Exchangeable Image File Format). Текстовое описание раздела EXIF-файла состоит из тегов, описывающих определенный параметр и значение этого параметра. Набор тегов содержит стандартизованную и обязательную часть, а также разделы, принадлежащие производителям техники и программного обеспечения для специальных целей. Программное обеспечение, позволяющее читать EXIF-данные, ставит в соответствие тегам их определения, а значениям – значения тегов. Хотя метаданные выдают некоторую информацию об изображении, они не всегда является корректными. Например, производители не всегда придерживаются спецификации, и потому случаются несовпадения тегов с их определениями. Также значения тегов легко могут быть изменены нарушителями, что останется незамеченным, поэтому определить подлинность изображений по метаданным невозможно [4].

Следующий способ, позволяющий определять отсутствие модификаций изображений, связан с компьютерной криминалистикой. Отдельный раздел данного направления посвящен целостности изображений и других данных. Для подтверждения неизменности данных используются однонаправленные хэш-функции. Эксперт, получив на исследование копию изображения, подсчитывает хэш-функцию некоторых ее атрибутов. Если ее значение совпадает со значением, внесенным в протокол, эксперт и иные лица получают определенную уверенность, что исследуемая копия совпадает с оригиналом [5].

Данный способ основан на том, что каждое изображение, полученное фотокамерой определенного производителя и модели, обладает свойствами, которые позволяют его ассоциировать именно с этим устройством. К этим свойствам можно отнести настройки, содержащиеся в устройстве и используемые для формирования изображения – таблица Хаффмана, размер изображения, таблица квантования. Кроме того, сюда же можно отнести способ задания метаданных, который отражается в структуре EXIF каждого полученного изображения на данном фотоаппарате. Далее по значениям этих параметров подсчитывается хэш-функция, создается база таких хэш-значений. После этого каждая новая фотография сравнивается с тем, что есть в базе – наличие такого же хэша говорит о принадлежности изображения какому-либо классу, ассоциированному с определенной моделью фотокамеры. Минус данного подхода – в том, что необходимо всегда иметь оригинал для создания базы данных хэшей.

В табл. 1 представлено сравнение характеристик современных методов с разработанным.

	Разработанный алгоритм	Error Level Analysis [6]	Color Adjustment [7]
Возможность анализа данных на уровне пикселей	отсутствует	присутствует	присутствует
Простота реализации	да	нет	нет
Обобщение метода на все виды редактирования изображений	Определяет любые изменения: дубликация элемента изображения, удаление части изображения, наложение и др.	Позволяет определить только наложенные части изображения	Позволяет определить дублированные элементы

Таблица 1. Сравнительная характеристика методов

Разработка метода

Спецификация формата JPEG. Согласно спецификации T.81 [8] формат JPEG состоит из упорядоченного набора параметров и маркеров, описывающих сжатые данные [9]. Параметры и маркеры, в свою очередь, образуют сегменты. Маркеры служат для идентификации различных структурных частей формата JPEG. Большинство маркеров начинают сегмент маркера, содержащий связанную группу параметров; некоторые маркеры стоят в одиночку. Всем маркерам назначаются двухбайтовые коды: байт 0xFF обязательно должен стоять за байтом, который не равен 0 или 0xFF. Второй байт указывается для каждого определенного маркера (табл. 1). Согласно спецификации T.81 [8], маркеры, которые описывают структуру JPEG-изображения, не могут включать в себя подмаркеры. Структура типичного маркера представлена на рисунке.

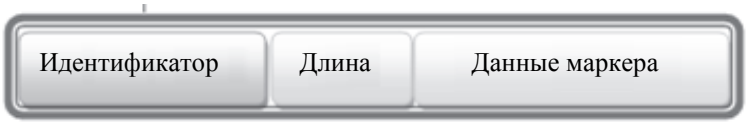

Рисунок. Структура маркера

Идентификатором являются два байта, обязательно в формате 0xFFC0, по которым можно идентифицировать тип маркера. Длина, как и идентификатор, состоит из двух байт, значение которых складывается из длины самой секции и длины данных маркера в байтах (в обратном порядке). Нужно отметить, что не все маркеры имеют длину (например, маркеры TEM, RST0…RST7, SOI, EOI). Данные маркера – набор байт, которые требуют обработки в соответствии с типом маркера, не могут включать в себя подмаркеры.

Все существующие маркеры условно можно разделить на три группы: основные, второстепенные и остальные. Это деление происходит по признаку встречаемости в изображениях формата JPEG. В табл. 2 приведены маркеры, которые встречаются в любом изображении JPEG.

Тип маркера	Идентификатор	Обозначение стандарта	Определение
SOF_0	$C0_{16}$	Baseline DCT	Начало кадра, базовый метод
SOF_1	$C1_{16}$	Extended sequential DCT	Начало кадра, расширенный, последовательный метод
SOF_2	$C2_{16}$	Progressive DCT	Начало кадра, прогрессивный метод
DHT	$C4_{16}$	Define Huffman table(s)	Определение таблиц Хаффмана
SOI	$D8_{16}$	Start of image	Начало изображения
EOI	$D9_{16}$	End of image	Конец изображения
SOS	DA_{16}	Start of scan	Начало скана
DQT	DB_{16}	Define quantization table(s)	Определение таблиц квантования

Таблица 2. Основные маркеры

В табл. 3 представлены все маркеры, которые хоть и встречаются так же часто, как и основные, но не требуют обязательной обработки для получения изображения [10].

Тип маркера	Идентификатор	Обозначение стандарта	Определение
RST_0–RST_7	$D0_{16}$–$D7_{16}$	Restart marker number 0–7	Определение интервала перезапуска от 0 до 7
DNL	DC_{16}	Define number of lines	Определение числа линий
DRI	DD_{16}	Define restart interval	Определение интервала перезапуска
COM	FE_{16}	Comment	Комментарий
APP_0–APP_{15}	$E0_{16}$–EF_{16}	Reserved for application segments	Зарезервированная информация о приложениях–упаковщиках

Таблица 3. Необязательные маркеры

Машинное обучение – это подраздел искусственного интеллекта, который рассматривает методы построения алгоритмов, способных обучаться. Главным различием методов является тип обучения: с учителем или без. В данной работе применяется первая группа методов. В отличие от существующих методов определения неизменности изображений (визуальный, по метаданным изображения), в результате машинного обучения метод способен выдавать результат на основе обученных параметров [11].

Для разработки метода автором был выбран язык программирования – Python, версия 2.7. Этот язык гибок и прост в использовании, а также имеет много возможностей для разработчиков. Выбранный язык программирования имеет много специализированных готовых библиотек с реализованными алгоритмами машинного обучения, поэтому подходит для разработки прототипа модели и проверки выдвинутых гипотез. Python имеет специальную библиотеку Scikit-Learn, в которой реализовано большое количество алгоритмов машинного обучения [12]. Для разработки метода автором использовались следующие алгоритмы: логистическая регрессия, наивный байесовский классификатор, *k*-ближайших соседей [13], деревья решений [14], метод опорных векторов [15], random forest [16].

Для реализации разработанного метода использовался IPython Notebook и следующая конфигурация оборудования: Intel i7–2600K, GeForce GTX 560, 10 GB RAM.

В общем случае алгоритм машинного обучения выглядит следующим образом.

Дано: X – множество изображений, Y – множество классов.

$\{x_1..x_n\} \subset X$ – набор тренировочных данных;

$y_i = y(x_i)$, $i = 1..n$ – известные ответы;

$F_j: X \to D_j, j=1..m$ – признаки объектов;

вектор $(\mathbf{f_1(x)},..,\mathbf{f_m(x)})$ – вектор характеристик.

Модель предсказания:

$A = \{g(x, \theta) \mid \theta \in \Theta\}$, где $g: X \times Q \to Y$ – фиксированная функция, Θ – множество допустимых значений параметра θ.

Метод обучения:

$\mu: (X \times Y)^n \to A$, которое произвольной выборке $x^n = (x_i, y_i)^n_{i=1}$ ставит в соответствие некоторый алгоритм $a \in A$.

Этап обучения:

метод μ по выборке $X^n = (x_i, y_i)^n_{i=1}$ строит алгоритм $a = \mu(X^n)$.

Этап тестирования:

алгоритм a для новых объектов x'_i выдает ответы $a(x'_i)$.

В качестве базового метода был реализован подход, похожий на существующие способы определения подлинности изображений. Его принцип заключается в создании базы хэшей таблиц квантования и таблиц кодов Хаффмана. На тренировочной выборке устанавливается, в каких классах встречался данный хэш. А на тестовой находится пересечение этих множеств:

- если остается один элемент, то можно точно установить класс и проверить правильность распознавания класса;
- если пересечение – пустое множество, то невозможно сделать вывод о принадлежности к какому-либо классу;
- если в пересечении несколько элементов, то также невозможно сделать вывод.

Очевидный минус такого подхода – необходимость создания большой базы данных. Он также является негибким – определение подлинности изображения марки, которой в базе нет, невозможно.

Выборка оригинальных изображений была получена путем скачивания с сайта www.steves-digicams.com. Здесь представлены различные модели фотоаппаратов вместе с примерами оригинальных фотографий.

С помощью специальной библиотеки httplib2 для Python было скачано 12518 фотографий. Изначально было 22 марки и около 200 моделей.

Полученные данные нуждались в предобработке: была необходима проверка на предмет испорченных изображений. Чтобы выборка была более сбалансированной, было решено использовать только те марки фотокамер, количество примеров фотографий которых больше 200. В итоге количество изображений сократилось до 11431 фотографий.

В качестве классов редакторов изображений были выбраны следующие программы: Adobe Photoshop CC 2015, Open CV2, Fotor, Pixlr, GIMP, ImageMagick, PIL, XnConvert. Как отдельный класс были также взяты изображения, сохраненные из социальных сетей (Facebook, Instagram, Vkontakte, Twitter, Telegram) и сгенерированные в редакторе Open CV2.

Выборка программ редактирования охватывает различные операционные системы: Windows (Adobe Photoshop CC 2015, GIMP, XnConvert); Unix (ImageMagick, PIL); OS X (Open CV2).

Также их можно разделить по критерию типа программы: консольные редакторы (ImageMagick, PIL, Open CV2); онлайн-редакторы (Fotor, Pixlr); графические редакторы (Adobe Photoshop CC 2015,

GIMP, XnConvert). Исходя из этого, можно утверждать, что выборка редакторов включает в себя все наиболее популярные редакторы изображений.

Выборка отредактированных изображений была получена из исходной выборки с помощью следующего алгоритма.

1. Генерация массива из 1000 случайных оригинальных фотографий.
2. Применение функций редактора с различными параметрами.
3. Сохранение отредактированных изображений.

Класс Social Media был получен путем скачивания изображений из социальных сетей – Facebook, Twitter, Instagram, Vkontakte, Telegram.

На основе изучения спецификации формата JPEG было решено использовать особенности структуры таких файлов в качестве признаков классификации, а именно – последовательность, количество маркеров, длину их секции.

Для начала в качестве основных признаков классификации были выбраны десять наиболее важных, по мнению авторов, характеристик: количество всех маркеров, количество маркеров начала 0xFFD8, длина и количество таблиц квантования 0xFFDB, длина и количество начала кадра, базового метода 0xFFC0, длина и количество таблиц кодов Хаффмана 0xFFC4, длина и количество начала закодированного изображения 0xFFDA. Все используемые маркеры являются основными для JPEG-изображений.

Для улучшения полученного результата было решено расширить количество характеристик до 43: основные маркеры формата JPEG – 0xFFC0, 0xFFC1, 0xFFC4, 0xFFD8, 0xFFD9, 0xFFDA, 0xFFDB; второстепенные маркеры – 0xFFD0..D7, 0xFFDC, 0xFFDD, 0xFFFE, 0xFFE1, 0xFFEC, 0xFFEE; а также для наглядности был взят единственный маркер из группы остальных (1.3) – 0xFFDE, который в 90% случаев не встречается в файлах JPEG и не требует обработки.

Следующим этапом в улучшении качества результата стало выделение 256 + 43 характеристик, где 43 – характеристики, используемые ранее, а 256 – вектор, характеризующий количество встречаемости каждого конкретного байта в первой таблице квантования JPEG-изображения. Данный способ имеет незначительное отличие от базового метода, но дает прирост значения результата.

Апробация метода и получение результатов

Для обучения модели необходимо обработать полученную выборку. Алгоритм обработки изображения включает в себя поиск заданных маркеров в байтовом представлении файла и преобразование полученной информации в нужный вид.

Для тестирования предложенных методов вся выборка случайно разделяется на тренировочные (70%) и тестовые (30%) данные.

После обработки изображений тестируются различные методы машинного обучения: логистическая регрессия – метод LogisticRegression из библиотеки sklearn.linear_model, а также функции expected и predicted (одинаковые для всех методов); наивный байессовский классификатор – метод GaussianNB из библиотеки sklearn.naive_bayes; *k*-ближайших соседей (*k*-nearest neighbors algorithm, *k*-NN) – метод KNeighborsClassifier из библиотеки sklearn.neighbors; деревья решений – метод DecisionTreeClassifier из библиотеки sklearn.tree; метод опорных векторов – метод SVC (Support Vector Classification) из библиотеки sklearn.svm; random forest – метод RandomForestClassifier из библиотеки sklearn.ensemble.

Результаты можно разделить на две категории. Первая – бинарная классификация, при которой модель разделяет все данные на два класса – оригинальный и модифицированный. Второй вид результата – многоклассовая классификация. Здесь же все данные разделяются на 20 классов – 10 классов моделей камер и 10 различных редакторов. Модель определяет, было ли изображение модифицировано и каким именно инструментом, либо же не изменялось совсем, и к какой модели фотокамеры оно принадлежит.

Итоговые результаты при бинарной классификации представлены в табл. 4. Курсивом выделены два алгоритма, которые показали наилучшие результаты.

Исходя из табл. 4, можно сделать вывод о том, что метод SVC дольше остальных обучается и тестируется, а наивный байесовский классификатор имеет наименьшие показатели затрат по времени, но наихудшие результаты. Наилучшие же методы – деревья решений и random forest – тратят в 2 раза больше времени на обучение при 299 признаках, но на тестировании затраты примерно равны. Поскольку обучение проходит единожды, показательным является только время, затраченное на тестирование.

Результаты, представленные в таблицах, говорят, во-первых, о том, что данные хорошо разделимы, а, во-вторых, что метод, предложенный авторами, может быть применен на практике.

Результаты трех лучших методов – random forest, деревья решений и *k*-ближайших соседей, демонстрируют прирост в процентах при использовании 299 признаков для классификации. В табл. 5 приведен сравнительный анализ результатов использования различного количества характеристик. Как видно из таблицы, максимальный прирост на многоклассовой классификации составляет 6,21% у метода деревья решений; на бинарной – 0,26% у того же метода.

№ п/п	Метод	Количество характеристик	Точность на обучающей выборке, %	Точность на тестовой выборке, %	Время на обучение, с	Время на тестирование, с
1	*Random forest*	*43*	*99,87*	*98,67*	*1,04*	*0,20*
2	*Random forest*	*299*	*99,86*	*98,48*	*2,09*	*0,37*
3	Наивный байессовский классификатор	43	61,81	61,51	0,89	0,19
4	Наивный байессовский классификатор	299	46,52	45,62	1,73	0,36
5	*Дерево решений*	*43*	*100*	*98,33*	*1,02*	*0,20*
6	*Дерево решений*	*299*	*100*	*98,35*	*2,33*	*0,36*
7	Логистическая регрессия	43	81,01	80,98	1,86	0,22
8	Логистическая регрессия	299	81,78	80,79	5,04	0,36
9	*k*-ближайших соседей	43	98,11	96,27	5,01	1,91
10	*k*-ближайших соседей	299	98,08	96,47	25,21	10,75
11	SVC	43	93,62	93,33	46,50	1,78
12	SVC	299	89,71	89,75	263,95	13,82

Таблица 4. Результаты бинарной классификации.
Курсивом выделены алгоритмы, показавшие наилучшие результаты

Количество характеристик	Результат метода *k*-NN, %		Результат метода деревья решений, %		Результат метода random forest, %	
	20 классов	2 класса	20 классов	2 класса	20 классов	2 класса
43	84,51	96,61	88,55	98,21	90,24	98,76
299	84,87	96,68	94,76	98,47	93,28	98,37

Таблица 5. Сравнительная таблица использования 43 и 299 признаков

№ п/п	Метод	Количество характеристик	Точность на обучающей выборке, %	Точность на тестовой выборке, %	Время на обучение, с	Время на тестирование, с
1	*Random forest*	*43*	*99,25*	*90,03*	*1,12*	*0,21*
2	*Random forest*	*299*	*97,61*	*93,15*	*2,04*	*0,48*
3	Наивный байессовский классификатор	43	53,61	52,52	1,04	0,23
4	Наивный байессовский классификатор	299	51,45	50,17	2,59	0,76
5	*Дерево решений*	*43*	*100*	*88,88*	*1,15*	*0,20*
6	*Дерево решений*	*299*	*100*	*94,71*	*2,45*	*0,36*
7	Логистическая регрессия	43	67,31	67,47	16,57	0,24
8	Логистическая регрессия	299	67,82	66,37	86,29	0,37
9	*k*-ближайших соседей	43	91,33	94,54	5,04	1,89
10	*k*-ближайших соседей	299	91,52	83,72	25,31	10,71
11	SVC	43	77,55	75,37	50,00	5,95
12	SVC	299	67,63	66,97	308,63	32,49

Таблица 6. Результаты многоклассовой классификации.
Курсивом выделены методы, показавшие наилучшие результаты

Исходя из полученных результатов, можно сделать вывод о том, что использование 299 признаков для классификации разумно. Полученный прирост имеет важное значение на практике.

Итоговая многоклассовая классификация представлена в табл. 6. Курсивом выделены методы, показавшие наилучшие результаты. Как видно из табл. 6, метод SVC по-прежнему обучается и тестируется дольше остальных, а наивный байесовский классификатор имеет наименьшие показатели затрат по времени на обучение. Однако, если основываться на критерии времени, затраченного на тестирование, наилучшими методами для многоклассовой классификации являются деревья решений и random forest.

Заключение

Определение нарушений целостности изображений является актуальной проблемой в области информационной безопасности. Несмотря на стремительное развитие информационных технологий, существующие методы решения данной задачи недостаточно хороши и требуют новых подходов.

Один из таких подходов рассматривается в данной работе. В ходе работы была получена выборка изображений 10 различных марок фотокамер и 10 графических редакторов. В качестве уникальных характеристик изображения, используемых для классификации, было предложено использовать структуру формата JPEG, а именно, длину и количество маркеров. Также было протестировано шесть методов машинного обучения, используемых для классификации. При сравнении полученных результатов разными методами был выбран наилучший – дерево решений.

В результате проделанной работы был получен метод, позволяющий определять модификации изображений с вероятностью 98,47%, а также марки фотокамер с вероятностью 94,71%. Такие высокие результаты и такие преимущества перед другими методами, как возможность работы при отсутствии базы оригинальных изображений, невозможность изменения исходных параметров изображения, использующихся в методе, и высокая скорость работы, говорят об эффективности разработанного метода, а также о возможности его практического применения. Характеристики, которые используются для классификации, содержатся в структуре JPEG-изображения, и не могут быть модифицированы специально, т.е. с их помощью можно точно определить, было ли изображение модифицировано.

Данный метод также имеет дальнейшие пути развития – например, пополнение базы данных оригинальных фотографий, расширение количества классов как марок, так и редакторов, поиск новых характеристик для классификации.

Литература

1. Хатунцев Н.А., Лизоркин А.М. Метод доказывания неизменности фотоизображений в рамках компьютерно-технической экспертизы (на примере из экспертной практики) // Теория и практика судебной экспертизы. 2014. № 3. С. 69–73.
2. Farid H. Digital Image Ballistics from JPEG Quantization: A Followup Study. Technical Report TR2008-638. Dartmouth College, 2008, 6 p.
3. Photo Tampering throughout History [Электронный ресурс]. Режим доступа: http://pth.izitru.com/1994_02_00.html, своб. (дата обращения: 20.10.2017).
4. Щербаков С. Метаданные в цифровой фотографии [Электронный ресурс]. 2005. Режим доступа: www.ixbt.com/digimage/metadxph.shtml, своб. (дата обращения: 20.10.2017).
5. Федотов Н.Н. Форензика – компьютерная криминалистика. М.: Юридический Мир, 2007. 432 с.
6. Beck T. How to Detect Image Manipulations? [Электронный ресурс]. 2017. Режим доступа: headt.eu/detect-image-manipulations, своб. (дата обращения: 30.10.2017).
7. Color Adjustment: HSV colorspace [Электронный ресурс]. Режим доступа: fotoforensics.com/tutorial-coloradjustment.php (дата обращения: 30.10.2017).
8. Recommendation T.81. Information technology – digital compression and coding of continuous-tone still images. Part 1. Requirements and guidelines. CCITT, 1993. 186 p.
9. Ватолин Д.С. Алгоритмы сжатия изображений. М.: МГУ, 1999. 76 с.
10. Шелепов М.И. История создания, устройство, строение и применение графического формата JPEG [Электронный ресурс]. Режим доступа: www.kolpinkurs.ru/stati/jpeg.htm, своб. (дата обращения: 25.10.2017).
11. Domingos P. A few useful things to know about machine learning // Communications of the ACM. 2012. V. 55. N 10. P. 78–87. doi: 10.1145/2347736.2347755
12. Pedregosa F., Varoquaux G., Gramfort A. et al. Scikit-learn: machine learning in python // Journal of Machine Learning Research. 2011. V. 12. P. 2825–2830.
13. Cai Y.I., Ji D., Cai D.F. A KNN research paper classification

References

1. Khatuntsev N.A., Lizorkin A.M. Method of proof invariably images in the computer-technical expertise (from expert practice). *Theory and Practice of Forensic Science*, 2014, no. 3, pp. 69–73. (In Russian)
2. Farid H. Digital Image Ballistics from JPEG Quantization: A Followup Study. *Technical Report TR2008-638*. Dartmouth College, 2008, 6 p.
3. *Photo Tampering throughout History*. Available at: http://pth.izitru.com/1994_02_00.html (accessed: 20.10.2017).
4. Shcherbakov S. *Metadata in Digital Photography*. 2005. Available at: www.ixbt.com/digimage/metadxph.shtml (accessed: 20.10.2017).
5. Fedotov N.N. *Forensic - Computer Criminology*. Moscow, Yuridicheskii Mir Publ., 2007, 432 p. (In Russian)
6. Beck T. *How to Detect Image Manipulations*? 2017. Available at: headt.eu/detect-image-manipulations (accessed: 30.10.2017).
7. *Color Adjustment: HSV colorspace*. Available at: fotoforensics.com/tutorial-coloradjustment.php (accessed: 30.10.2017).
8. *Recommendation T.81. Information technology – digital compression and coding of continuous-tone still images. Part 1. Requirements and guidelines*. CCITT, 1993, 186 p.
9. Vatolin D.S. *Image Compression Algorithms*. Moscow, MSU Publ., 1999, 76 p. (In Russian)
10. Shelepov M.I. *History of creation, organisation, structure and use of JPEG graphic format*. Available at: www.kolpinkurs.ru/stati/jpeg.htm (accessed: 25.10.2017).
11. Domingos P. A few useful things to know about machine learning. *Communications of the ACM*, 2012, vol. 55, no. 10, pp. 78–87. doi: 10.1145/2347736.2347755
12. Pedregosa F., Varoquaux G., Gramfort A. et al. Scikit-learn: machine learning in python. *Journal of Machine Learning Research*, 2011, vol. 12, pp. 2825–2830.
13. Cai Y.I., Ji D., Cai D.F. A KNN research paper classification method based on shared nearest neighbor. *Proc. NTCIR-8 Workshop Meeting*. Tokyo, 2010, 5 p.
14. Quinlan J.R. Induction of decision trees. *Machine Learning*, 1986, vol. 1, no. 1, pp. 81–106. doi:

method based on shared nearest neighbor // Proc. NTCIR-8 Workshop Meeting. Tokyo, 2010. 5 p.
14. Quinlan J.R. Induction of decision trees // Machine Learning. 1986. V. 1. N 1. P. 81–106. doi: 10.1023/A:1022643204877
15. Hsu C.W., Chang C.C., Lin C.J. A Practical Guide to Support Vector Classification. [Электронный ресурс]. 2016. URL: www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf (дата обращения: 25.10.2017).
16. Ali J., Khan R., Ahmad N., Maqsood I. Random forests and decision trees // IJCSI International Journal of Computer Science. 2012. V. 9. N 5. P. 272–278.

10.1023/A:1022643204877
15. Hsu C.W., Chang C.C., Lin C.J. *A Practical Guide to Support Vector Classification*. 2016. URL: www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf (accessed: 25.10.2017).
16. Ali J., Khan R., Ahmad N., Maqsood I. Random forests and decision trees. *IJCSI International Journal of Computer Science*, 2012, vol. 9, no. 5, pp. 272–278.

Авторы

Серова Алиса Игоревна – инженер, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, ORCID ID: 0000-0002-7667-6520, Aliceinwobderland25@gmail.com

Спивак Антон Игоревич – кандидат технических наук, доцент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, Scopus ID: 56779715800, ORCID ID: 0000-0002-6981-8754, Anton.spivak@gmail.com

Authors

Alice I. Serova – engineer, ITMO University, Saint Petersburg, 197101, Russian Federation, ORCID ID: 0000-0002-7667-6520, Aliceinwobderland25@gmail.com

Anton I. Spivak – PhD, Associate Professor, ITMO University, Saint Petersburg, 197101, Russian Federation, Scopus ID: 56779715800, ORCID ID: 0000-0002-6981-8754, Anton.spivak@gmail.com