Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ”

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

«Определение подлинности JPEG изображений с учетом аппаратно-программных характеристик цифровых фотокамер»

Автор Серова Алиса Игоревна _____
(Фамилия, Имя, Отчество) (Подпись)

Направление подготовки (специальность) 10.03.01

Информационная безопасность

Квалификация бакалавр
(бакалавр, инженер, магистр)

Руководитель Сорокин И.В., к.т.н. _____
(Фамилия, И., О., ученое звание, степень) (Подпись)

К защите допустить

Зав. кафедрой Зикратов И.А.,профессор,д.т.н. _____
(Фамилия, И., О., ученое звание, степень) (Подпись)

“___”_____ 20 ____г.

Санкт-Петербург, 2016 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В настоящее время информационные технологии находят все более широкое применение в самых различных сферах деятельности людей. А фотографии в свою очередь становятся неотъемлемой частью общения, но широкая доступность легких инструментов для редактирования изображений ставит под сомнение их подлинность, когда это очень важно. В связи с этим количество различных способов мошенничества постоянно растет. К примеру, в процессе страховании автомобиля делается ряд фотографий, отображающих состояние машины на момент страхования. Будучи измененными злоумышленниками в будущем, такие фотографии уже не будут нести достоверную информацию. Современные цифровые фотокамеры и фотографии, полученные с их помощью, являются носителями криминалистически значимой компьютерной информации [9].

Если нужно решить была ли фотография отредактирована с помощью специальных программ или иным способом, проводится исследование предоставленных файлов, снятых с помощью фотоаппарата. Данное исследование, конечно, может быть проведено путем визуального изучения изображения на предмет наличия в нем каких-либо изменений. Но различные эксперты, проводящие исследование, могут не сойтись во мнении. Поэтому очень важно уметь различать оригинальную фотографию от измененной автоматически, например, методами машинного обучения. Эта проблема является актуальной в области информационной безопасности и криминалистики.

Алгоритм JPEG в наибольшей степени пригоден для сжатия фотографий и картин, содержащих реалистичные сцены с плавными переходами яркости и цвета [11]. Наибольшее распространение JPEG получил в цифровой фотографии и для хранения и передачи изображений с

использованием сети Интернет. Поэтому было решено рассматривать изображения именно в этом формате.

Целью данной работы является разработка модели для распознавания правдоподобности JPEG изображения и определение его источника (фотокамера или графический редактор) при помощи методов машинного обучения.

Для достижения цели были поставлены и решены следующие задачи:

- подготовка выборки, состоящей из примеров фотоснимков, сделанных с помощью различных моделей фотокамер;
- подготовка выборки, состоящей из изображений, отредактированных в различных графических редакторах;
- выделение основных характеристик JPEG изображений для классификации;
- применение различных методов машинного обучения для классификации изображений.

Объектом исследования является подлинность изображений в формате JPEG, а предметом исследования – программно-аппаратные характеристики цифровых фотокамер.

Теоретической и методологической основой исследования послужили различные методы машинного обучения, а также исследования других ученых в данной области.

В работе использовались научные труды российских и зарубежных ученых, учебно-методические материалы, ресурсы Интернет.

Практическая значимость ВКР заключается в том, что разработанная модель может применяться для подтверждения подлинности JPEG изображений компаниями в различных сферах деятельности: журналистика, юридические или страховые фирмы, онлайн аукционы и др.

В первой главе рассматривается структура JPEG-изображений в виде маркеров и параметров, происходит разделение всех маркеров на группы по степени важности. Вторая глава посвящена обзору существующих методов, включающих визуальное определение подлинности изображений, извлечение информации из метаданных и обзор специальных тестов компьютерной криминалистики.

В третьей главе описываются различные методы машинного обучения, которые используются в практической части, их преимущества и недостатки. А в четвертой главе рассказывается экспериментальная часть работы: поэтапное решение каждой из поставленных задач, получение и объяснение результатов.

1 ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

1.1 Спецификация формата JPEG

Согласно спецификации T.81 [18] формат JPEG состоит из упорядоченного набора параметров и маркеров, описывающих сжатые данные [7]. Параметры и маркеры в свою очередь образуют сегменты.

Параметры представляют собой целые числа, со значениями, характерными для процесса кодирования, характеристиками исходного изображения, а также другими особенностями, выбранными приложениями. Параметрам присваиваются значения либо 4 - бит, 1 байт, или 2 - байтовые коды. За исключением некоторых дополнительных групп параметров, есть группы параметров кодирования критически важной информации, без которых невозможно корректное декодирование изображения.

Присвоение кода для параметра должно быть целым числом без знака, указанной длины в битах с конкретным значением параметра.

Для параметров длиной 2 байта (16 бит) наиболее значимый байт должен быть на первом месте в упорядоченной последовательности байтов сжатых данных. Параметры длиной в 4 бита всегда стоят парами, и такая пара должна быть закодирована в один байт. Первый 4 - битовый параметр пары должны занимать наиболее значимые 4 бита байта. В любой 16- , 8- или 4 - битовый параметр, старший значащий бит должен стоять первым, а младший значащий бит - в последнюю очередь.

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 (см. табл.1). Согласно спецификации T.81 [18] маркеры, которые описывают структуру JPEG-изображения, не могут включать в себя подмаркеры.

Структура типичного маркера представлена на Рисунке 1.

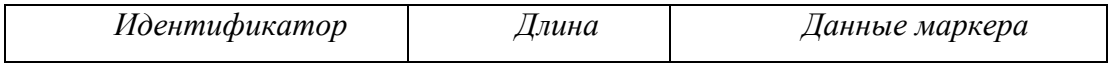

Рисунок 1 – Структура маркера

Идентификатором являются два байта, обязательно в формате 0xFF*C0*, по которым можно идентифицировать тип маркера.

Длина так же, как и идентификатор состоит из двух байт, значение которых складывается из длины данной секции (2) 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.

1.2 Основные маркеры из формата JPEG

Все существующие маркеры условно можно разделить на три группы: основные, второстепенные и остальные. Это деление происходит по признаку встречаемости в изображениях формата JPEG. В таблице 1 приведены маркеры, которые встречаются в любом изображении JPEG.

Таблица 1 – Основные маркеры

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

Маркер SOF_0

Наличие данного маркера говорит о том, что изображение закодировано базовым методом, который является самым распространенным. Базовый метод ограничен по количеству определяемых Таблиц Хаффмана – по два каждого типа (AC или DC).

Маркер SOF_1

Наличие такого маркера показывает то, что изображение закодировано так называемым расширенным последовательным методом. Этот процесс идентичен процессу кодирования базовым методом, за исключением того, количества используемых наборов Таблиц Хаффмана, число которых увеличивается до четырех. Четыре Таблицы Хаффмана DC и четыре Таблицы Хаффмана AC являются максимально допустимым количеством по спецификации T81.

Маркер SOF_2

Данный маркер указывает на то, что изображение закодировано прогрессивным методом. При нем сначала загружается изображение в низком разрешении, а потом и сама картинка. Понять, что изображено, можно не дожидаясь окончательной загрузки изображения. Данный метод является не менее популярный, чем базовый и позволяет определять до четырех Таблиц Хаффмана каждого типа, как и расширенный последовательный метод.

Маркер DHT

Маркер DHT (Define Huffman Table – Определение Таблицы Хаффмана) определяет (или переопределяет) таблицы Хаффмана. Сами Таблицы имеют два важных параметра – количество и тип (AC или DC). Один маркер DHT может определять множество таблиц, однако общее количество каждого типа ограничено применяемым методом. Единственное ограничение на размещение маркеров DHT состоит в том, что если скан требует наличия особого идентификатора таблицы или типа, идентификатор

должен быть определен ранее в файле маркером DHT. Структура данного маркера представлена в таблице 2.

Таблица 2 - Структура маркера DHT

Размер поля	Описание
1 байт	4 старших бита определяют тип таблицы. Значение 0 указывает на таблицу DC, значение 1 указывает на таблицу AC. 4 младших бита определяют идентификатор таблицы. Значение этого идентификатора равно 0 или 1, для кадров базового метода и 0, 1, 2 или 3 для кадров прогрессивного и расширенного методов.
16 байт	Счетчик кодов Хаффмана длиной от 1 до 16. Каждый счетчик хранится в 1 байте.
Переменный	1-байтовые символы, сортированные по коду Хаффмана. Число символов равно сумме 16 счетчиков кода.

Все Таблицы Хаффмана представляют собой последовательность из фиксированных 17 байтов данных и поле переменной длины, содержащее до 256 дополнительных байтов, следующих за ними. 16 байтов образуют массив из 1-байтовых целых чисел без знака, элементы которого задают число кодов Хаффмана для каждой возможной длины кода (1—16). Сумма из 16 длин кодов представляет собой число значений в таблице Хаффмана. Каждое значение равно 1 байту и за ним следуют, согласно порядку кода Хаффмана, счетчики длины. Число таблиц Хаффмана, указываемое маркером DHT, определяется из поля длины.

Маркер SOI

Маркер SOI (Start of Image – Начало изображения) говорит о начале JPEG-изображения. Такой маркер может быть только один, и он находится в самом начале файла. Также он стоит в одиночку.

Маркер EOI

Маркер EOI (End of Image – Конец изображения) показывает конец JPEG-изображения и всегда находится в конце JPEG-изображения. А это значит, что после данного маркера не могут находиться другие маркеры.

Аналогично предыдущему маркеру SOI, маркер EOI всегда единственный и стоит в одиночку.

Маркер SOS

Данный маркер SOS (Start of Scan) говорит о начале секции закодированного изображения. В файле маркер обязательно должен появляться после маркера SOF_N. Также условием использования маркера является то, что ему должны предшествовать все маркеры DHT и DQT, которые определяют Таблицы Хаффмана и таблицы квантования, используемые в скане.

Маркер DQT

Маркер DQT (Define Quantization Table – Определение таблиц квантования) определяет (или переопределяет) таблицы квантования, используемые в изображении. Данный маркер может определять несколько таблиц квантования – до четырех, в зависимости от используемого метода. Определение таблиц квантования следует за полем длины маркера. Значение поля длины состоит из суммы размеров таблиц и еще двух байт, которые и занимает само поле длины. Структура маркера представлена в таблице 3.

Таблица 3 – Структура маркера DQT

Размер поля	Описание
1 байт	4 младших бита являются идентификатором таблиц квантования таблицы (возможные значения: 0, 1, 2 или 3). 4 старших бита задают размер значения квантования (0 – однобайтовые величины, 1– двухбайтовые величины).
64 или 128 байт	64 однобайтовых или двухбайтовых значений таблицы квантования без знака.

Каждая таблица квантования начинается с 1-го байта, который содержит информацию о таблице. В зависимости от того, какие значения имеют 4 старших бита информационного байта, каждое значение таблицы квантования равно 1 либо 2 байтам, а длина всего определения таблицы составляет 1+64 либо 1+128 байтов. Двухбайтовые значения квантования

могут использоваться только с 12-битовыми дискретизованными данными. 4 младших бита определяют числовой идентификатор таблицы. За информационным байтом следуют 64 значения квантования, которые хранятся в JPEG-изображении в зигзагообразном порядке, как показано на рисунке 2.

Рисунок 2 – Пример хранения значений квантования

В файле изображения маркеры DQT могут появляться в любой части. Единственное ограничение состоит в том, что если скану требуется таблица дискретизации, она должна быть определена в предшествующем маркере DQT.

1.3 Второстепенные маркеры из формата JPEG

В таблице 4 представлены все маркеры, которые хоть и встречаются так же часто, как и основные, но не требуют обязательной обработки для получения изображения.

Таблица 4 – Необязательные маркеры

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

Маркеры RST_N

RST_0…RST_7 являются маркерами перезапуска и используются для маркировки блоков независимо кодированных сжатых данных скана. В этих маркерах отсутствуют поле длины и данные, а сами маркеры могут встречаться только внутри сжатых данных скана. Маркеры перезапуска могут использоваться для нейтрализации ошибок. Интервал между маркерами перезапуска определяется маркером DRI. Например, если интервал перезапуска равен нулю, то маркеры перезапуска не используются. В сканах маркеры перезапуска должны обязательно стоять по порядку RST_0, RST_1,...RST_7, RST_0...

Маркер DNL

Маркеры DNL используются для определения или переопределения размеров изображения внутри сжатых данных, а не внутри маркера SOF_n. Обычно маркеры DNL не используются, и большинство программ не могут их обрабатывать.

Маркер DRI

Маркер DRI (Define Restart Interval – Определение интервала перезапуска) задает число MCU’s между маркерами перезапуска внутри сжатых данных. Значение поля длиной 2 байта, следующего за маркером, всегда равно 4. В маркере есть только одно поле данных - 2-байтовое значение, которое определяет интервал перезапуска. Если значение интервала ноль - это означает, что маркеры перезапуска не используются. Маркер DRI с ненулевым значением интервала перезапуска может применяться для повторной активизации маркеров перезапуска далее в изображении.

Эти маркеры могут появляться в любом месте файла для определения или переопределения интервала перезапуска, который сохраняет свое действие до конца изображения или пока другой маркер DRI не изменит его. Чтобы маркеры перезапуска были включены в сегмент сжатых данных, в файле должен присутствовать где-нибудь маркер DRI.

Маркеры перезапуска необходимы для нейтрализации ошибок. Если декодер находит искаженные данные скана, он может использовать идентификатор маркера перезапуска и интервал перезапуска для определения места, с которого будет возобновлено декодирование изображения.

Маркер COM

Маркер COM (Comment - Комментарий) используется для хранения строк комментариев, например, информации об авторских правах. Их интерпретация зависит от конкретной программы, однако обычно что декодер игнорирует данную информацию. Именно маркер COM, а не маркер APP_N должен использоваться для хранения простого текста комментария. В файле JPEG он может появляться где угодно.

Маркеры APP_N

Маркеры APP_0…APP_{15} содержат специфические данные программы. Эти маркеры используются программами редактирования изображений для сохранения дополнительной информации, помимо той, что задается стандартом JPEG. Формат этих маркеров зависит от конкретной программы. Поле длины, располагающейся после маркера, может использоваться для пропуска данных маркера. За исключением маркеров APP_0, которые используются форматом JFIF, программа может игнорировать маркеры APP, которые она не распознает. Если программе необходимо сохранить информацию, выходящую за пределы возможностей форматов JPEG и JFIF, то для сохранения этой информации программа создает маркеры APP_N. Внутри файла JPEG маркеры APP_N могут появляться где угодно. По соглашению программы, которые создают маркеры APP_N, сохраняют свое имя (заканчивающееся нулем) в начале маркера, с целью предотвращения конфликтов с другими программами. Программа, обрабатывающая маркеры APP_N должна проверять не только идентификатор маркера, но также имя программы, записавшей маркер [10].

В таблице 5 представлены все оставшиеся маркеры, которые почти не встречаются в файлах JPEG и не требуют никакой обработки.

Таблица 5 – Остальные маркеры

Тип маркера	Иденти-фикатор	Обозначение стандарта	Определение
SOF_3	$C3_{16}$	Lossless (sequential)	Начало кадра, метод сжатия без потерь
SOF_5	$C5_{16}$	Differential sequential DCT	Начало кадра, дифференциальный последовательный метод
SOF_6	$C6_{16}$	Differential progressive DCT	Начало кадра, дифференциальный прогрессивный метод
SOF_7	$C7_{16}$	Differential lossless (sequential)	Начало кадра, дифференциальный метод без потерь
JPG	$C8_{16}$	Reserved for JPEG extensions	Резерв для последующих расширений формата JPEG
SOF_9	$C9_{16}$	Extended sequential DCT	Начало кадра, расширенный последовательный метод, арифметическое кодирование
SOF_{10}	CA_{16}	Progressive DCT	Начало кадра, прогрессивный метод, арифметическое кодирование
SOF_{11}	CB_{16}	Lossless (sequential)	Начало кадра, метод без потерь, арифметическое кодирование
SOF_{13}	CD_{16}	Differential sequential DCT	Начало кадра, дифференциальный последовательный метод, арифметическое кодирование
SOF_{14}	CE_{16}	Differential progressive DCT	Начало кадра, дифференциальный прогрессивный метод, арифметическое кодирование
SOF_{15}	CF_{16}	Differential lossless (sequential)	Начало кадра, дифференциальный метод без потерь, арифметическое кодирование
DAC	CC_{16}	Define arithmetic coding conditioning(s)	Определений условий арифметического кодирования
DHP	DE_{16}	Define hierarchical progression	Определение иерархической прогрессии
EXP	DF_{16}	Expand reference component(s)	Раскрытие справочных компонент
JPG_0…JPG_{13}	$F0_{16}$...FD_{16}	Reserved for JPEG extensions	Резерв для последующих расширений формата JPEG
TEM	01_{16}	For temporary private use in arithmetic coding	Для временного локального определения настроек арифметического кодирования
RES	02_{16}...BF_{16}	Reserved	Резерв для последующих расширений формата JPEG

Поскольку данные маркеры очень редко встречаются в JPEG-изображениях, автором было принято решение не рассматривать такие маркеры в отдельности.

2 СУЩЕСТВУЮЩИЕ РЕШЕНИЯ В ОБЛАСТИ ИССЛЕДОВАНИЯ

2.1 История развития фотографий. Визуальный метод

Проблема подлинности изображений появилась вместе с изобретением фотоаппаратов. Но если вначале можно было определять оригинальность изображений визуально, то с развитием технологий это становится все сложнее. Далее следует несколько примеров, показывающих историческое развитие подделок фотографий. Первым примером является известная фотография президента США - Авраама Линкольна. На самом деле данная фотография является подделкой и представляет из себя композицию головы Линкольна и тела Южного политика Дж. Калхуна. Фотография изображена на рисунке 3.

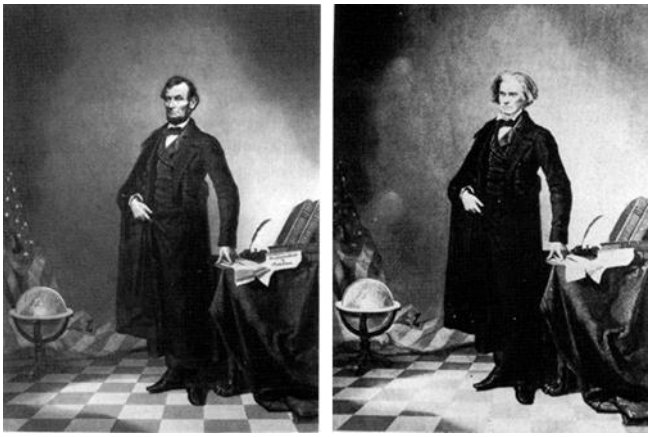

Рисунок 3 – Пример фото – подделки, датируемой 1860 г.

Следующий пример показывает, как с помощью подделок можно манипулировать новостями и обществом. В 1994 году появилась цифровая подделка: олимпийские конькобежцы Т. Хардинг и Н. Керриган появились на обложке New York Newsday – соперники были показаны репетирующими вместе, хотя на самом деле они являются соперниками [14]. Обложка показана на рисунке 4.

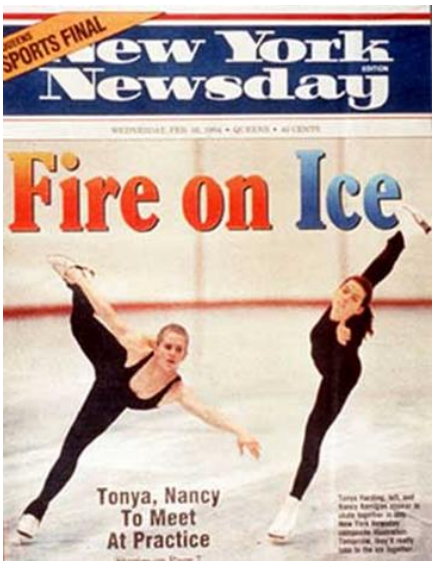

Рисунок 4 – Пример одной из первых цифровых фото-подделок, 1994 г.

С появлением цифровых фотоаппаратов (1981г.) и разработкой формата JPEG (1991г.), стали появляться различные методы определения подлинности изображений.

2.2 EXIF данные

Следующим этапом в определении подлинности фотографий стала проверка метаданных.

Формат EXIF (Exchangeble image format) необходим для корректной работы различных устройств с мультимедийными файлами. Данный формат предусматривает хранение данных изображения или музыки, их уменьшенных копий и подраздела текстового описания данных в одном файле.

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 их специальных целей. Программное обеспечение, позволяющее читать EXIF данные, ставит в соответствие тегам их определения, а значениям — значения тегов. Создатели не всегда придерживаются спецификации и

потому случаются несовпадения. Например, производитель камеры может записать в поле, соответствующее тегу, информацию в некорректном формате или вообще о другом параметре. Если это узкоспециализированная программа под определенную марку камер, то недоразумений обычно не бывает. Пользователи обязаны своей покупкой подчиняться описанным в руководстве ПО правилам. Но если это универсальная программа, то путаница с неправильным наименованием тегов и форматированием, соответствующих им величин, случается не редко [6].

Все вышесказанное говорит о том, что хоть метаданные выдают некоторую информацию об изображении, они не всегда является правильной. Также значения тегов легко изменить нарушителям, что останется незамеченным, поэтому определить подлинность изображений по метаданным возможно далеко не всегда.

2.3 Компьютерная криминалистика

Компьютерная криминалистика (форензика) - это прикладная наука о раскрытии и расследовании преступлений, связанных с компьютерной информацией, о методах получения и исследования доказательств, имеющих форму компьютерной информации (т.н. цифровых доказательств), о применяемых для этого технических средствах.

Отдельный раздел данной науки посвящен целостности изображений и других данных. Для удостоверения целостности и неизменности данных используются однонаправленные хэш-функции. Эксперт, получив на исследование копию, подсчитывает с нее хэш-функцию. Если ее значение совпадает со значением, внесенным в протокол, эксперт и иные лица получают уверенность, что исследуемая копия совпадает с оригиналом с точностью до бита [1].

Способы определения целостности JPEG-изображений постоянно совершенствуются. На сегодняшний момент у формата выделяются следующие признаки:

- метаданные EXIF;
- таблица Хаффмана, записанная в устройство заранее, которая определяется производителем на основании характеристик устройства;
- таблицы квантования, которые также записаны в устройство заранее и определяются производителем;
- размеры эскиза, характерные для устройств определенной марки и модели. При этом эскиз также имеет характерные для используемого устройства эскизные таблицы Хаффмана и Квантования;
- размер итогового изображения;

То есть фотоизображение, созданное фотокамерой определенной марки и модели, должно соответствовать характерным для этой фотокамеры размеру изображения, структуре EXIF, таблице Хаффмана изображения, таблице Квантования изображения, размеру эскиза, таблице Хаффмана эскиза, таблице Квантования эскиза.

Далее по этим параметрам подсчитывается хэш-функция, создается база таких хэшей. После чего каждая новая фотография сравнивается с тем, что есть в базе – наличие такого же хэша говорит о принадлежности изображения какому-либо классу.

Также существует ряд методов по работе с любыми форматами изображений на уровне пикселей. Более глубокий анализ включает в себя анализ специфики сжатия фотографий, анализ артефактов, появившихся вследствие редактирования определенных частей изображений, а также анализ частей фотографий, которые были скопированы и изменены [12]. Данные методы хоть и являются универсальными, с другой стороны сложны в реализации, подвержены к переобучению и, как правило, разрабатываются на определение какого-то конкретного способа редактирования изображения. В данной работе такие методы не рассматриваются.

Существующие сервисы, использующие в качестве методов определения подлинности изображений вышеназванные методы:

1) www.izitru.com;

Данный сервис собирает базу оригинальных фотографий с помощью пользователей. Для проверки подлинности использует комбинацию шести тестов судебно-медицинской экспертизы [13].

2) www.smtdp.com/pmi

Сервис позволяет определять наличие редактирования. Для определения используются различные характеристики JPEG изображений в комплексе, например, анализ таблиц квантования, эффект двойного квантования – было ли изображение сохранено в формате JPEG более одного раза, наличие факта копирования изображения и другие. Данный сервис имеет два продукта. Первый – PMI (picture manipulation inspector),

Второй продукт – DD (dublicate detector) предлагает распознавать дубликаты изображений. Для этого создаются уникальные отпечатки особых частей изображений, устойчивых к незначительным преобразованиям, и сохраняются в базу данных. При поиске похожих изображений отпечатки предоставленного изображения сравниваются с отпечатками в имеющейся базе данных, и при определенном пороге схожести возвращаются наиболее значимые результаты [17].

3) forensicservices.ru

Лаборатория цифровой форензики - российская компания, предоставляющая услуги в области ИБ для бизнеса. Ее деятельность сводится к трем направлениям: фото и видеотехническая экспертиза, компьютерно-техническая экспертиза и расследование инцидентов ИБ [8]. Что касается первого направления, компания решает следующие задачи:

- идентификация изображений (установка соответствия между цифровым устройством и цифровым изображением), для проведения нужен доступ к устройству или тестовые данные с устройства;

- идентификация оригинальности изображения: определение оригинальности изображения, проверка подлинности фотографий, определение методов обработки примененных к изображению при его не оригинальности;
- идентификация ретуши и монтажа: нахождение или констатация отсутствия следов изменения конкретных областей в графияеских редакторах;
- исследование метаданных: определение характеристик, условий съемки, времени, даты последней оцифровки и других характеристик изображения по метаданным.

2.4 Описание базового метода

В качестве базового подхода автором был реализован метод, 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 – можно точно установить класс и проверить правильность распознавания класса;
- если пересечение пустое множество – 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 не гибким – определение подлинности изображения марки, которой в базе нет – невозможно.

Результаты практической реализации описаны в главе 4.4 и будут сравниваться с остальными методами.

3. ПРИМЕНЕНИЕ МАШИННОГО ОБУЧЕНИЯ

3.1 Целесообразность использования

Машинное обучение — это подраздел искусственного интеллекта, который рассматривает методы построения алгоритмов, способных обучаться. Главным различием методов является тип обучения: с учителем или без [5]. В данной работе применяется первая группа методов. В отличие от существующих методов определения правдоподобности изображений (визуальный, по метаданным изображения), в результате машинного обучения модель способна обобщать результаты и выдавать их на основе обученных параметров. Некоторые методы машинного обучения способны обучаться на относительно небольшой выборке данных, однако, чем больше будет база для обучения, тем точнее будет полученный результат.

3.2 Описание используемых методов

Для разработки модели автором был выбран язык программирования – Python, версия 2.7. Этот язык гибкий, простой в использовании, а также имеет много возможностей для разработчиков.

Python имеет специальную библиотеку Scikit-Learn, в которой реализовано большое количество алгоритмов машинного обучения [15]. Ниже представлено описание тех алгоритмов, которые использовались автором для разработки модели.

1. Логистическая регрессия.

Данный метод считается классическим и обычно используется для решения задач бинарной классификации, но допускается и многоклассовая [2]. Логистическая регрессия полезна тогда, когда необходимо предсказать наличие или отсутствие характеристики или итога на основании значений набора переменных. Логистическая регрессия является наиболее простым

методом, а минус в том, что она чувствительна к большим одиночным выбросам данных.

2. Наивный байесовский классификатор.

Также является одним из классических алгоритмов машинного обучения, основанный на применении Теоремы Байесса. Основной задачей данного метода является восстановление плотностей распределения данных обучающей выборки [2]. Достоинством такого классификатора является малое количество данных, необходимых для оценки параметров, требуемых для классификации. Часто этот метод дает хорошее качество в задачах многоклассовой классификации.

3. K-ближайших соседей.

Метод kNN (k-Nearest Neighbors) часто используется как составная часть более сложного алгоритма классификации. Например, его оценку можно использовать как признак для обьекта. Иногда простой kNN на хорошо подобранных признаках дает отличное качество [2].

4. Деревья решений.

Classification and Regression Trees (CART) часто используются в задачах, в которых обьекты имеют категориальные признаки и используется для задач регресии и классификации. Очень хорошо деревья подходят для многоклассовой классификации. Достоинствами являются быстрое построение модели и легкая интерпретируемость, а недостатком является то, что этот метод часто сходится на локальном решении [3]. Также немаловажным достоинством является возможность построения графика важности применяемых признаков, например, на основе критерия информационного прироста.

5. Метод опорных векторов.

SVM (Support Vector Machines) - набор контролируемых методов обучения, используемых для классификации, регрессии. Изначально применялись для задач бинарной классификации, но есть способы, позволяющие использовать его и для мультиклассификации - методом one-

vs-all. Существенным недостатком методов является то, что они чувствительны к шумам и стандартизации, а преимуществом можно считать то, что эти методы наиболее быстрый для нахождения решающих функций [4]. В данной работе используется метод SVC (Support Vector Classification) из группы методов SVM [16].

6. Random forest.

Метод, придуманный после CART, в основе которого лежит использование ансамбля деревьев принятия решений.

Суть алгоритма заключается в случайной выборке переменных на каждой итерации и построении на этой новой выборке построение дерева принятия решений. При этом производится “bagging” — выборка случайных двух третей наблюдений для обучения, а оставшаяся треть используется для оценки результата. Такую операцию проделывают тысячи раз. Результирующая модель будет результатом “голосования” набора полученных при моделировании деревьев. Достоинством метода является высокое качество результата, особенно для данных с большим количеством переменных и малым количеством наблюдений, возможность распараллеливания, необязательное наличие тестовой выборки. А к недостаткам можно отнести то, что модель для достижения хороших результатов должна проделать как можно большее количество итераций, в результате чего модель становится большой – и ее интерпретация усложняется [3].

4 ПРАКТИЧЕСКАЯ ЧАСТЬ И РЕЗУЛЬТАТЫ

4.1 Подготовка выборки оригинальных фотографий

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. Далее, для того, 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. Распределение фотографий по классам показано в таблице 6.

Таблица 6 - Распределение фотографий по классам

№п/п	Марка	Количество фотографий
1	Canon	3017
2	Casio	407
3	Fujifilm	1212
4	Kodak	238
5	Nikon	1779
6	Olympus	1523
7	Panasonic	699
8	Pentax	558
9	Samsung	1053
10	Sony	945

4.2 Подготовка выборки редактированных изображений

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

Выборка программ редактирования охватывает различные ОС:

- 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);
- ПО (Adobe Photoshop CC 2015, GIMP, XnConvert).

Исходя из этого, можно утверждать, что выборка редакторов охватывает различные возможные стороны редактирования 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 SocialMedia был получен путем скачивания изображений из социальных сетей – Facebook, Twitter, Instagram, Vkontakte, Telegram. В таблице 7 изображена сводная таблица всех редакторов.

Таблица 7 - Сводная таблица графических редакторов

№п/п	Редактор	Количество фотографий
1	GIMP	1000
2	ImageMagick	1000
3	Fotor	1000
4	Open CV2	1000
5	Open CV2 (generate)	1000
6	PIL	1000
7	Pixlr	1000
8	Photoshop CC	1000
9	SocialMedia	1000
10	XnConvert	1000

4.3 Формирование признаков для классификации

В ходе изучения спецификации формата JPEG (Глава 1.1), автором была предложена инновационная идея использовать особенности структуры таких файлов в качестве признаков классификации, а именно: последовательность, количество маркеров и их длину секции.

Вначале в качестве основных признаков классификации были выбраны десять наиболее важных по мнению автора 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 (Глава 1.2).

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 никакой 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 схож с базовым методом и дает прирост значения результата (Глава 4.3).

4.4 Обучение модели 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 данные (70%) и тестовые (30%).

Результаты базового метода:

- при классификации выборки, состоящей только из оригинальных изображений, данный метод дает результат в 52%;
- при добавлении в модель еще 10 классов, точность данного метода значительно уменьшается – до 17%. И даже при бинарной классификации точность составляет всего лишь 21%.

Столь плохие результаты связаны с частым повтором таблиц квантования и таблиц кодов Хаффмана среди различных классов изображений. Это требует выделение дополнительных параметров и совершенствование такой модели. С подробными результатами обработки изображений можно ознакомиться в таблице 8.

Таблица 8 - Результаты базового метода

	Многоклассовая классификация		Бинарная классификация	
	Кол-во изображений	%	Кол-во изображений	%
Правильно	1133	17,29	1424	21,73
Неправильно	191	2,91	159	2,42
Ничего не нашлось	83	1,26	83	1,26
Больше одного класса	5145	78,52	4886	74,57

Следующим этапом является тестирование методов машинного обучения для классификации изображений.

Прежде всего, для всех данных необходимо получить набор из 43 или 43+256 характеристик.

После получения набора данные были стандартизированы при помощи метода preprocessing из библиотеки sklearn:

standardized_X = preprocessing.scale(vyb_dataset_x)

Далее можно применять различные методы машинного обучения.

1. Логистическая регрессия.

Для реализации данной модели использовался метод LogisticRegression из библиотеки sklearn.linear_model, а также функции expected и predicted (они одинаковы для всех методов, поэтому их упоминание в дальнейшем будет опущено).

Результаты данного метода (Таблица 9) - средние, для улучшения следует использовать другой метод.

Таблица 9 - Результаты обучения с помощью логистической регрессии

	Тренировочные данные, %		Тестовые данные, %	
	43 характ.	43+256 хар-к	43 характ.	43+256 хар-к
Многоклассовая классификация	67,98	68,61	67,86	66,57
Бинарная классификация	81,203	81,07	81,95	80,99

2. Наивный байессовский классификатор.

Для реализации данной модели использовался метод GaussianNB из библиотеки sklearn.naive_bayes.

Данный метод показал наихудшие результаты. Результаты показаны в таблице 10.

Таблица 10 - Результаты обучения с помощью наивного Байеса

	Тренировочные данные, %		Тестовые данные, %	
	43 характ.	43+256 хар-к	43 характ.	43+256 хар-к
Многоклассовая классификация	53,25	51,01	53,19	50,22
Бинарная классификация	61,52	46,32	61,68	46,12

3. K-ближайших соседей.

Для реализации данной модели использовался метод KNeighborsClassifier из библиотеки sklearn.neighbors.

Результаты показаны в таблице 11. Они оказались лучше результатов первых двух методов.

Таблица 11 - Результаты обучения с помощью метода k-ближайших соседей

	Тренировочные данные, %		Тестовые данные, %	
	43 характ.	43+256 хар-к	43 характ.	43+256 хар-к
Многоклассовая классификация	91,18	91,28	84,51	84,87
Бинарная классификация	97,91	98,01	96,61	96,68

4. Деревья решений.

Для реализации данной модели использовался метод DecisionTreeClassifier из библиотеки sklearn.tree.

Результаты данного метода показаны в таблице 12. Они оказались одними из лучших.

Таблица 12 - Результаты обучения с помощью деревьев решений

	Тренировочные данные, %		Тестовые данные, %	
	43 характ.	43+256 хар-к	43 характ.	43+256 хар-к
Многоклассовая классификация	99,99	100	88,55	94,76
Бинарная классификация	100	100	98,21	98,47

Данный метод позволяет построить график важности используемых характеристик на основе индекса Джини (Gini). Для 43 характеристик график показан на рисунке 5, а для 43 + 256 – на рисунке 6.

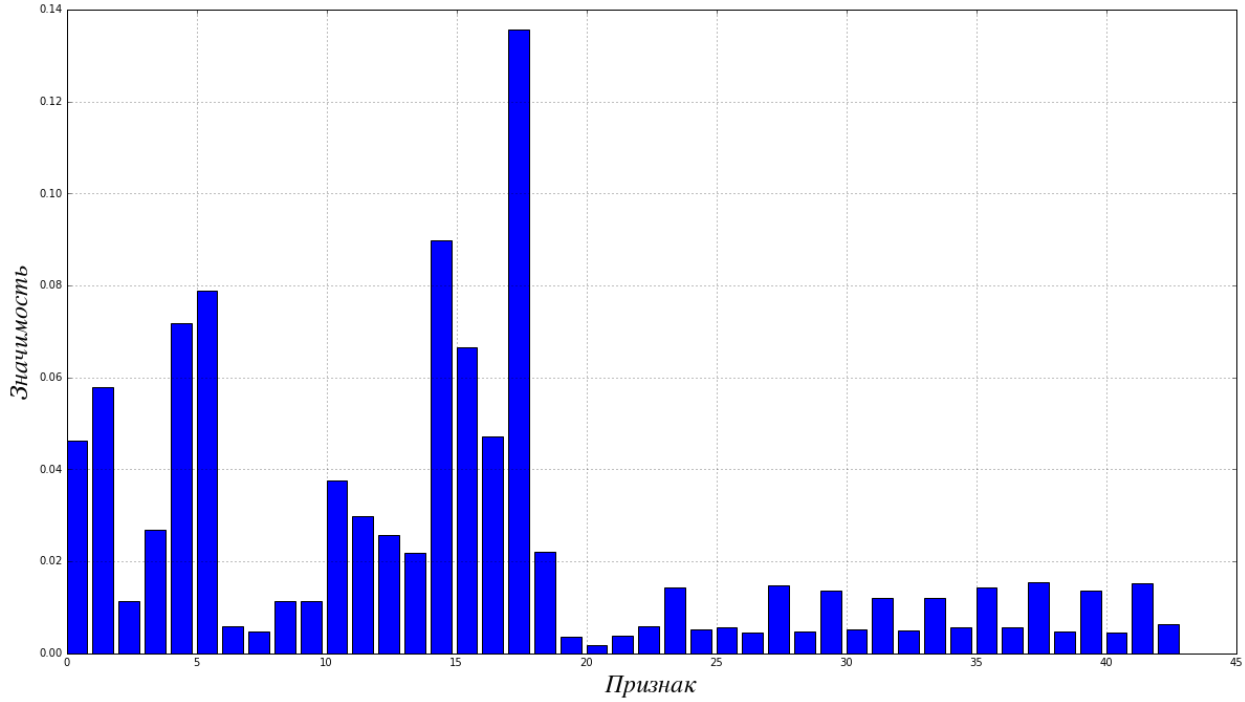

Рисунок 5 – Значимость каждого из 43 признаков, полученная с использованием Дерева решений

Как видно из графика, 17 признак имеет самую большую значимость. Это маркер xFFD7, который не относится к группе основных маркеров. Следующий по важности маркер - xFFD7, из той же группы, что и предыдущий. И третий по важности – xFFDB. Этот маркер уже относится к группе основных. Однако каждый из выбранных автором признаков влияет на классификацию в той или иной степени.

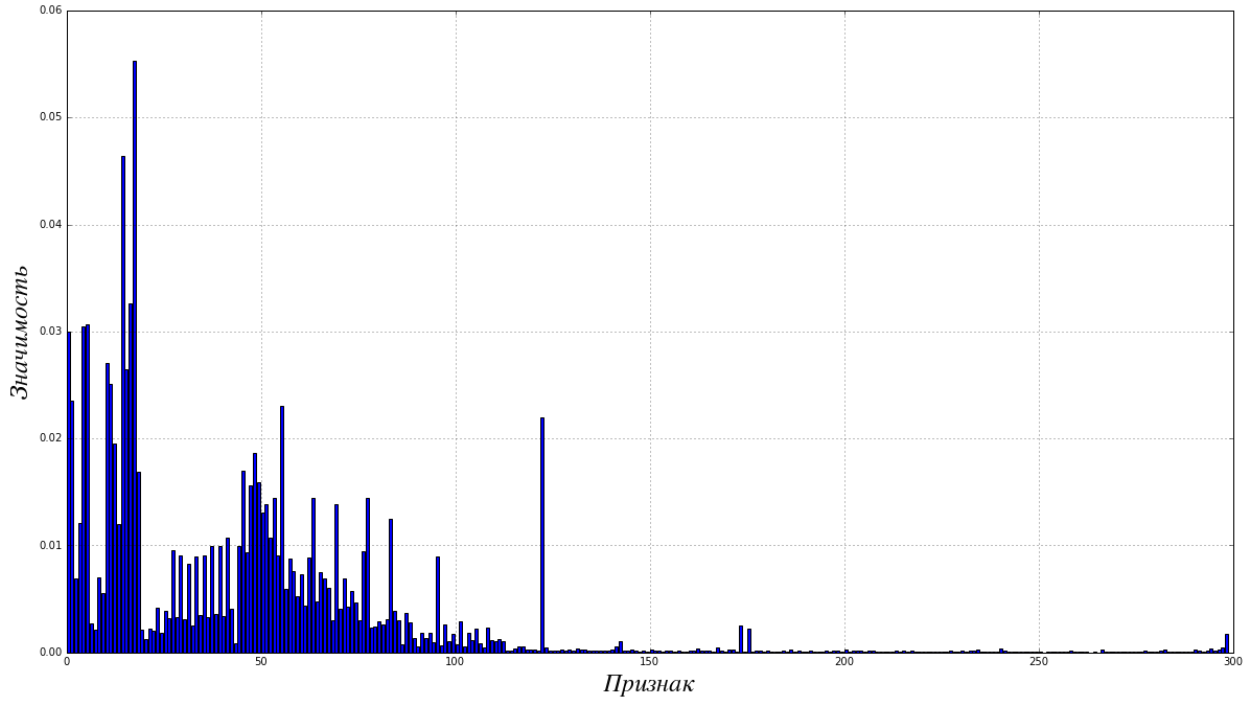

Рисунок 6 – Значимость каждого из 299 признаков, полученная с использованием Дерева решений

Данный график интересен тем, что он показывает значимость остальных 256 характеристик. Несмотря на то, что большинство байт не оказывают существенного влияния на классификацию, учет этого вектора дает результат на 5 процентов больше. А если собрать еще большую базу изображений, то значимость может возрасти.

5. Метод опорных векторов.

Для реализации данной модели использовался метод SVC из библиотеки sklearn.svm.

Результаты данного метода показаны в таблице 13. Его особенностью являются значительные затраты на расчеты вместе со средними значениями результатов.

Таблица 13 - Результаты обучения с помощью метода опорных векторов

	Тренировочные данные, %		Тестовые данные, %	
	43 характ.	43+256 хар-к	43 характ.	43+256 хар-к
Многоклассовая классификация	77,49	67,00	76,53	64,83
Бинарная классификация	93,31	89,63	93,75	89,36

6. Random forest.

Для реализации данной модели использовался метод RandomForestClassifier из библиотеки sklearn.ensemble.

Результаты данного метода показаны в таблице 14. Он показал самые лучшие результаты. Однако, из-за особенностей метода, было решено не использовать его результаты как наилучшие.

Таблица 14 - Результаты обучения с помощью метода Random forest

	Тренировочные данные, %		Тестовые данные, %	
	43 характ.	43+256 хар-к	43 характ.	43+256 хар-к
Многоклассовая классификация	99,28	99,44	90,24	95,15
Бинарная классификация	99,88	99,72	98,76	98,45

Далее в таблице 15 показана точность классификации на тестовой выборке для каждого класса марки фотокамер.

Таблица 15 – Итоговая точность по маркам фотокамер

№п/п	Класс	Правильно угадано	Всего файлов в классе	Точность, %
1	Canon	882	884	99,77
2	Casio	111	115	96,52
3	Fujifilm	381	381	100
4	Kodak	67	67	100
5	Nikon	543	549	98,9
6	Olympus	456	471	94,96
7	Panasonic	193	193	100
8	Pentax	168	171	98,24
9	Samsung	332	337	98,51
10	Sony	283	298	94,96

Точность классификации по каждому классу больше 90%, а классы Fujifilm, Kodak и Panasonic имеют максимальную точность – 100%.

В таблице 16 показана точность классификации для различных классов – редакторов.

Таблица 16 – Итоговая точность по редакторам

№п/п	Класс	Правильно угадано	Всего файлов в классе	Точность, %
1	GIMP	323	328	98,47
2	ImageMagick	311	326	95,39
3	Fotor	297	297	100
4	Open CV2	306	306	100
5	Open CV2 (generate)	294	294	100
6	PIL	293	293	100
7	Pixlr	292	292	100
8	Photoshop CC	318	320	99,37
9	SocialMedia	293	293	100
10	XnConvert	295	303	97,35

Как видно из таблицы, точность классификации по каждому классу больше 90%, а у 6 классов точность наивысшая – 100%.

Результаты, представленные в таблицах 15 и 16 говорят, во-первых, о том, что данные хорошо разделимы, а, во-вторых, о том, что метод, предложенный автором, может быть применен на практике.

Используя результаты трех лучших методов – Random Forest, Деревья решений и К-ближайших соседей, покажем прирост в процентах при использовании 299 признаков для классификации. В таблице 17 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.

Таблица 17 – Сравнительная таблица использования 43 и 299 признаков

Кол-во характеристик	kNN, %		Деревья решений, %		Random Forest, %	
	20 кл.	2 кл.	20 кл.	2 кл.	20 кл.	2 кл.
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

Еще одним аспектом, заслуживающим внимание, является учет времени, потраченного на работу каждого алгоритма. На первый взгляд расширение количества характеристик до 299 должно затруднить подсчет результатов. Однако это не совсем так. В таблице 18 сравниваются затраты каждого из методов на один файл при бинарной классификации.

Таблица 18 – Сравнение затрат по времени при бинарной классификации

№ п/п	Метод	Кол-во хар-к	Точность на обучающей выборке, %	Точность на тестовой выборке, %	Время на обучение, с	Время на тест-е, 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

Исходя из данной таблицы, можно сделать вывод о том, что метод SVC дольше остальных обучается и тестируется, а наивный байессовский классификатор имеет наименьшие показатели затрат по времени. Наилучшие методы - деревья решений и Random Forest затрачивают времени в два раза больше на обучение при 200 признаках, но на тестировании затраты примерно равны. Но поскольку обучение проходит единожды, показательным является время, затраченное на тестирование.

Ниже, в таблице 19, отображены затраты времени при многоклассовой классификации. Они схожи с предыдущими результатами.

Таблица 19 – Сравнение затрат по времени при многоклассовой класс-ии

№п/п	Метод	Кол-во хар-к	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

ЗАКЛЮЧЕНИЕ

Определение подлинности изображений является актуальной проблемой в области информационной безопасности. Несмотря на стремительное развитие информационных технологий, существующие методы решения данной проблемы 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 была получена модель, позволяющая определять правдоподобность изображений с вероятностью 98,45%, а также марки фотокамер с вероятностью 95,15%.

Такой результат позволяет найти практическое применение в области информационной безопасности и криминалистике.

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

Таким образом, задачи решены в полном объеме, а цель достигнута – модель, разработанная при помощи методов машинного обучения способна определять правдоподобность JPEG изображения и определять его источник с высокой степенью вероятности.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Федотов Н.Н. Форензика – компьютерная криминалистика – М.: Юридический Мир, 2007. – 432 с.: ил. ISBN 5-91159-015-8

2. Введение в машинное обучение с помощью Python и Scikit-Learn. [Электронный ресурс] / URL: https://habrahabr.ru/company/mlclass/blog/247751 Режим доступа: свободный. (дата обращения: 23.04.2016)

3. Классификация и регрессия с помощью деревьев принятия решений. [Электронный ресурс] / URL: https://habrahabr.ru/post/116385 Режим доступа: свободный. (дата обращения: 25.04.2016)

4. Машина опорных векторов. [Электронный ресурс] / URL: http://www.machinelearning.ru/wiki/index.php?title=SVM Режим доступа: свободный. (дата обращения: 25.04.2016)

5. Машинное обучение. [Электронный ресурс] / URL: http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение Режим доступа: свободный. (дата обращения: 26.04.2016)

6. Метаданные в цифровой фотографии. [Электронный ресурс] / URL: http://www.ixbt.com/digimage/metadxph.shtml Режим доступа: свободный. (дата обращения: 20.04.2016)

7. Сжатие данных. Алгоритмы и форматы. [Электронный ресурс] / URL: http://www.proximasp.ru/compression_of_data.html Режим доступа: свободный. (дата обращения: 15.04.2016)

8. Фото и видеотехническая экспертиза. [Электронный ресурс] / URL: http://forensicservices.ru/expertise/Photo_Video Режим доступа: свободный. (дата обращения: 4.05.2016)

9. Хатунцев Н. А. Метод доказывания неизменности фотоизображений в рамках компьютерно-технической экспертизы // Научное сообщество студентов XXI столетия. ГУМАНИТАРНЫЕ НАУКИ: сб. ст. по мат. XX междунар. студ. науч.-практ. конф. № 5(20) [Электронный ресурс]/ URL:

http://sibac.info/archive/guman/5(20).pdf Режим доступа: свободный. (дата обращения: 11.04.2016)

10. Шелепов М.И. История создания, устройство, строение и применение графического формата JPEG.[Электронный ресурс] / URL: http://www.kolpinkurs.ru/stati/jpeg.htm Режим доступа: свободный. (дата обращения: 15.04.2016)

11. H. Farid. Digital Image Ballistics from JPEG Quantization: A Followup Study. TR2008-638, Department of Computer Science, Dartmouth College, September 2008. — 6 с.

12. FourAndSix. [Электронный ресурс] / URL: http://www.fourandsix.com/ fourmatch Режим доступа: свободный. (дата обращения: 4.05.2016)

13. IZITRU About us. [Электронный ресурс] / URL: http://www.izitru.com/aboutus.php Режим доступа: свободный. (дата обращения: 4.05.2016)

14. Photo Tampering throughout history. [Электронный ресурс] / URL: http://pth.izitru.com/1994_02_00.html Режим доступа: свободный. (дата обращения: 4.05.2016)

15. Scikit-learn. [Электронный ресурс] / URL: http://scikit-learn.org/stable/ Режим доступа: свободный. (дата обращения: 19.04.2016)

16. Sklearn.svm.SVC. [Электронный ресурс] / URL: http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC Режим доступа: свободный. (дата обращения: 27.04.2016)

17. System of methods & tools of digital processing. [Электронный ресурс]/ URL: http://www.smtdp.com/pmi Режим доступа: свободный. (дата обращения: 4.05.2016)

18. Recommendation T.81. Information technology – digital compression and coding of continuous-tone still images requirements and guidelines. — The International telegraph and telephone consultative committee, 1993. — 186 с.