DOI: 10.15514/ISPRAS–2020–32(4)–15

Использование доменно-состязательного обучения для распознавания текстовых капч

[1] *Кущук Д.О., ORCID: 0000-0002-8778-8608 <dkuschuk@ispras.ru>*
[2] *Рындин М.А., ORCID: 0000-0002-7504-3975 <mxrynd@ispras.ru>*
[2] *Яцков А.К., ORCID: 0000-0002-1312-1675 <yatskov@ispras.ru>*
[2] *Варламов М.И., ORCID: 0000-0002-1083-6210 <varlamov@ispras.ru>*
[1] *Московский физико-технический институт*
141701, Россия, Московская область, г. Долгопрудный, Институтский пер., 9
[2] *Институт системного программирования им. В.П. Иванникова РАН,*
109004, Россия, г. Москва, ул. А. Солженицына, д. 25

Аннотация. Несмотря на появление более продвинутых вариантов публичных тестов Тьюринга, в настоящее время текстовая капча является достаточно распространённой, поэтому создание методов ее автоматического решения актуальны и сегодня. Современные алгоритмы успешно справляются с этой задачей, однако, обладают рядом ограничений, таких как: неспособность работать с изменяющейся длиной текста на изображении, медленное и сложное обучение. В данной работе представлен алгоритм атак на текстовые капчи, не требующий априорного знания длины текста на изображении. Экспериментально показано, что использование данного алгоритма совместно с методом состязательного обучения позволяет добиваться высокого качества на реальных данных, используя 200-500 размеченных примеров для обучения. Экспериментальное сравнение разработанного метода с современными аналогами показало, что при использовании одинакового числа реальных примеров для обучения наш алгоритм показывает сравнимое или более высокое качество, при этом он имеет более высокую скорость работы и обучения.

Ключевые слова: машинное обучение; решение капчи; OCR; состязательное обучение

Для цитирования: Кущук Д.О., Рындин М.А., Яцков А.К., Варламов М.И. Использование доменно-состязательного обучения для распознавания текстовых капч. Труды ИСП РАН, том 32, вып. 4, 2020 г., стр. 203–216. DOI: 10.15514/ISPRAS–2020–32(4)–15

Using Domain Adversarial Learning for Text Captchas Recognition

[1] *Kushchuk D.O., ORCID: 0000-0002-8778-8608 <dkuschuk@ispras.ru>*
[2] *Ryndin M.A., ORCID: 0000-0002-7504-3975 <mxrynd@ispras.ru>*
[2] *Yatskov A.K., ORCID: 0000-0002-1312-1675 <yatskov@ispras.ru>*
[2] *Varlamov M.I., ORCID: 0000-0002-1083-6210 <varlamov@ispras.ru>*
[1] *Moscow Institute of Physics and Technology*
9 Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russian Federation
[1] *Ivannikov Institute for System Programming of the RAS,*
25, Alexander Solzhenitsyn Str., Moscow, 109004, Russia

Abstract. Nowadays the problem of legal regulation of automatic collection of information from sites is being actively solved. This means that interest in tools and programs for automatic data collection is growing and that's why interest in automatic programs for solving CAPTCHA («Completely Automated Public Turing test to tell Computers and Humans Apart») is increasing too. In spite of creation of more advanced types of captcha, nowadays text captcha is quite common. For instance, such large services as Yandex, Google, Wikipedia, VK

continue to use them. There are many methods of breaking text captchas in literature, however, it should be noted that most of them have a limitation to priori know the length of the text on the image, which is not always the case in the real world. Also, many methods are multi-stage, which brings additional inconvenience to their implementation and application. Moreover, some methods use a large number of labeled pictures for training, but in reality, to collect data one has to be able to solve captchas from different sites. Respectively, manually labeling dozens of thousands of examples for training for each new type of captcha is an unrealistically difficult action. In this paper we propose a one-step algorithm of attack on text captchas. This approach does not require a priori knowledge of the text's length on the image. It has been shown experimentally that the usage of this algorithm in conjunction with the adversarial learning method allows one to achieve high quality on real data, using the low number (200-500) of labeled examples for training. An experimental comparison of the developed method with modern analogs showed that using the same number of real examples for training, our algorithm shows a comparable or higher quality, while it has a higher speed of working and training.

Keywords: machine learning; captcha solving; OCR; adversarial learning

For citation: Kushchuk D.O., Ryndin M.A., Yatskov A.K., Varlamov M.I. Using Domain Adversarial Learning for Text Captchas Recognition. Trudy ISP RAN/Proc. ISP RAS, vol. 32, issue 4, 2020. pp. 203–216 (in Russian). DOI: 10.15514/ISPRAS–2020–32(4)–15

1. Введение

На сегодняшний день в мире происходит процесс правового урегулирования автоматического сбора информации с сайтов, а значит интерес к автоматическим программам решения капчи возрастает. Капча (Captcha) – это аббревиатура от фразы «Completely Automated Public Turing test to tell Computers and Humans Apart» (Полностью автоматизированный публичный тест Тьюринга, который отличает людей от компьютеров).

Благодаря автоматическому сбору информации исследователи и аналитики получают доступ к большим объемам актуальных данных для работы. Так как решение капчи вносит дополнительную латентность в процесс сбора, владельцы сайтов продолжают иметь дополнительный слой защиты от атаки на свои сервисы.

Так, например, в статье [1] во время краулинга сайта Яндекс.Новости, из-за большого количества запросов к серверу, авторы сталкивались с блокировкой сбора информации и были вынуждены распознавать капчу самостоятельно. Это создавало трудности для свободного сбора информации.

Несмотря на появление более продвинутых вариантов капчи, на сегодняшний день текстовая капча является достаточно распространённой. Например, такие крупные сервисы как Яндекс, Google, Wikipedia, Вконтакте продолжают использовать именно их (см. рис. 2). Текстовые капчи легки в разработке и привычны для пользователей. Поэтому задача развития методов распознавания и совершенствования текстовой капчи всё еще актуальна.

На данный момент существует множество методов распознавания текстовой капчи. Например, в статье [2] используется свёрточная сеть VGG, а в статье [3] используется предварительная сегментация картинки. Эти алгоритмы являются алгоритмами обучения с учителем и требуют большого объема размеченных обучающих данных, в то время как методы [4-6] основаны на генеративно-состязательном подходе и могут быть отнесены к методам обучения с частичным привлечением учителя. Данные алгоритмы состоят из двух основных фаз: в первой фазе происходит генерация примеров для обучения с использованием большого количества неразмеченных данных и, возможно, небольшого количества размеченных примеров. Во второй фазе осуществляется сам процесс обучения на полученном наборе данных. Этот подход позволяет значительно сократить количество примеров для обучения, что в задаче распознавания капч необходимо, ведь собирать и размечать реальные данные с сайтов достаточно трудоемкий процесс. Лучший в этой области метод представлен в статье [7], он достигает высокой точности на многих типах капч с различных сайтов. Однако, данный метод умеет работать только с капчами фиксированной длины.

Недавняя статья [8], в которой авторы используют идеи обучения представлениями, обходит ограничение априорного знания длины текста, но показывает более низкое качество распознавания. Метод доменно-состязательного обучения, в котором к основной модели добавляется дискриминатор [9], помогает устранить многофазность предыдущего подхода. Но в данной статье авторы используют основную архитектуру, способную к разгадыванию капч только с фиксированной длиной.

В данной работе мы представляем новый метод, совмещающий названные выше подходы и обладающий такими достоинствами как: простота архитектуры, точность, высокая скорость обучения и работы, требование малого числа примеров для обучения. Но главное преимущество – возможность работы алгоритма на капчах с произвольной длиной текста.

Во втором разделе данной работы будет приведен обзор современных методов. В первой части разд. 3 будет описан алгоритм распознавания текста на изображении. Во второй части представлен алгоритм DA-CRNN, основной идеей которого является использование состязательного обучения для снижения числа примеров, необходимого для обучения. В разд. 4 будет проведено экспериментальное тестирование разработанного метода.

2. Обзор существующих решений

2.1 Свёрточные нейронные сети

CNN (сверточная нейронная сеть) -- тип нейросетей, используемый для анализа изображений. Примеров сверточных архитектур множество – VGG [10], ResNet [11], LeNet-5 [12]. Логичным является применение сверточных архитектур для расшифровки шумных изображений. Авторы статьи [2] так и поступили, и использовали архитектуру, похожую на архитектуру VGG, для распознавания текстовой капчи.

Данный метод имеет ряд ограничений. Во-первых, длина текста на картинке должна быть строго фиксированной, для каждого вида капчи обучается своя модель. Например, в данной статье капча состоит из 5 символов. Во-вторых, все обучающие примеры должны быть размечены, авторы используют генератор капч BotDetect captcha [15] для генерации 100000 картинок для обучения и 1000 картинок для тестирования. В реальности, нужно уметь разгадывать капчи со сторонних сайтов, соответственно, вручную разметить 100000 примеров для обучения для каждого нового типа капчи является трудновыполнимым действием.

Похожие методы используются в и в других статьях с некоторыми дополнениями и особенностями. Например в [14] используется активное обучение CNN, то есть в процессе обучения модели она сама выбирает примеры, на которых продолжать обучение, и эксперты их размечают.

2.2 Сегментация и распознавание

Другая статья [3] иллюстрирует иной метод распознавания текстовой капчи. Идея состоит в том, чтобы сначала производить сегментацию картинки, а потом на основе этой сегментации предсказывать отдельные символы.

Весь алгоритм состоит из 4 частей – сегментации, слайсера, маркера и судьи. Сначала производится множественная сегментация, то есть картинка разделяется на части, отделяя каждый символ друг от друга, причем таких вариантов разделения несколько. После этого с помощью слайсера выбираются сегменты и строится граф, далее граф анализируется маркером и присваивается значение каждому сегменту, после чего судья выбирает самый вероятный ответ.

Так как задача множественной сегментации картинки (первые два слоя) в данном случае решается без учителя, то целесообразно использовать метод обучения с подкреплением. При

правильной и удачной сегментации, модель не спрашивает ничего человека, а при неправильной эксперту задается вопрос, в каком шаге была произведена ошибка: в сегментации или в распознавании? Именно так и происходит обучение модели.

2.3 Генеративные методы

В 2014 году была опубликована статья [15], в которой авторы представили генеративно-состязательные нейронные сети, в которых происходит обучение двух частей - дискриминатора и генератора. Такие сети в основном используются для генерации синтетических изображений, очень похожих на реальные.

В 2017 году опубликовалась статья [4], которая использует этот подход для распознавания текстовой капчи. В ней сначала генерировались искусственные картинки с заранее известным текстом, похожие на настоящие капчи. Далее на них обучалась модель распознавания текста на изображении. После чего получившуюся модель использовали для предсказания текста на реальных капчах. Тем самым исчезает необходимость собирать и вручную размечать большое количество реальных картинок для последующего обучения модели на них, ведь теперь на основе реальных неразмеченных изображений генерируются похожие на них искусственные размеченные изображения, которые и используются впоследствии для обучения.

Модель, которая использовалась для предсказания, состоит из нескольких сверточных слоев, после которых используются рекуррентные слои.

Также в 2017 году был опубликован метод [5], который полностью основан на генеративно-состязательном методе и решает задачу распознавания автомобильных номеров. А метод [6] решает задачу анализа фотографий уличных вывесок и знаков путем адаптации размеченных синтетических изображений на реальные уличные вывески. Причем в этой работе авторы используют не только попиксельную адаптацию пространства представления, но также и всю геометрию фотографий, углов, ракурсов.

После этого был опубликован метод расшифровки капчи [7], также основанный на GAN (генеративно-состязательная сеть), который на данный момент является одним из самым эффективных в задаче распознавания капчи с частичным привлечением учителя.

Данный метод, предложенный Гуйсинь Е (Guixin Ye) в 2018 году, представляет из себя четырехступенчатый метод. На первом шаге происходит генерация синтетических картинок, максимально похожих на реальные капчи. На втором шаге реализуется предобработка картинки – удаление шумов и чистка фона. На третьем шаге происходит обучение классификатора на этих искусственно созданных и очищенных капчах. Четвертый шаг – шаг улучшения качества, посредством частичного дообучения на небольшом количестве реальных размеченных изображений (производится тонкая настройка модели).

Данный метод является многоступенчатым, что усложняет обучение и отладку: может быть непонятно, какой из шагов вносит наибольшее ухудшение в работу общего алгоритма. Также стоит отметить один ощутимый недостаток – длина текста на изображении строго фиксирована. Авторы сами указывают на этот недостаток и предлагают сначала распознавать длину капчи, а уже потом использовать описанный выше метод. Однако это затрудняет обучение генератора, т.к. реальные изображения вначале придётся разметить и разделить по длине, что усложняет обучение и ведет к дополнительной работе экспертов по разметке.

2.4 Обучение представлениями

Относительно недавно была опубликована статья [16], которая демонстрирует вариант обучения представлениями CPC. Обучение представлениями – это техника, которая позволяет модели обнаруживать и выявлять внутренние признаки в неразмеченных данных. Иными словами, мы выявляем признаки в режиме самообучения, то есть используя для обучения сами целевые данные без их меток. Конкретно в данной модели CPC

использовалась идея выявления признаков путем обучения модели предсказывать следующие элементы последовательности, зная предыдущие. Таким образом и происходит отбор целевых признаков.

Авторы статьи [8] спроецировали данный метод на задачу выявления целевых признаков из реальных картинок. Их модель, названная ``extractor'', училась находить определенные сущности из входного изображения.

Далее к модели «extractor» добавлялся классификатор, и получившаяся модель с замороженными весами модели «extractor» тренировалась разгадывать обычные общедоступные размеченные синтетические капчи. Но, так как «extractor» изначально обучался выявлять признаки из целевых картинок, то модель добилась высокой точности классификации и на реальных изображениях.

Рассмотрены современные методы распознавания шумных картинок. Стоит отметить, что во многих способах существует ограничение в необходимости априорного знания длины текста на картинке, что в реальном мире не всегда так. Также многие методы являются многоступенчатыми, что привносит дополнительное неудобство в реализацию и применение методов. В следующем разделе мы попытаемся отойти от идеи многоступенчатости, а также устраним ограничение, касающееся фиксированной длины текста.

3. Построение решения задачи

3.1 Описание метода распознавания текста на изображении

В 2015 году вышла статья Баогуан Ши (Baoguang Shi), Сян Бай (Xiang Bai) и Конг Яо (Cong Yao) [17]. Основа их метода – нейронная сеть с архитектурой CRNN (Сверточная рекуррентная нейронная сеть). Архитектура такой нейронной сети состоит из трех частей - сверточных слоев, рекуррентных слоев и слоев транскрипции.

В начале сверточная часть извлекает последовательность признаков из каждого изображения. Далее эта последовательность признаков попадает в рекуррентные слои LSTM [18], после чего проходит слой softmax, где формируется предсказание для каждого элемента этой последовательности. После этого, эта предсказанная последовательность, образующаяся на выходе из рекуррентных слоев и слоя softmax, попадает в слой транскрипции, где преобразуется в последовательность готовых символов. Благодаря одноступенчатости этого метода, сеть тренируется, используя только одну общую функцию потерь. Для определения условной вероятности авторы используют CTC слой, описанный в [19]. Этот слой часто используется в задачах распознавания рукописного текста [20]. При этом для обучения необходимо знать только ответы к изображению, без уточнения, в какой части картинки находится тот или иной символ.

За основу нашего метода мы взяли рассмотренную архитектуру CRNN с параметрами, представленными в табл. 1.

Также при обучении в качестве оптимизирующего алгоритма использовался стохастический градиентный спуск [21] с изменяющимися в течение обучения параметрами. Параметр шага градиентного спуска (скорость обучения) μ изменялся на каждой эпохе по формулам:

$$\mu = \frac{0{,}01}{(1 + 10 \cdot p)^{0.75}}, \text{где } p = \frac{\text{номер эпохи}}{\text{общее количество эпох}}. \quad (1)$$

Табл. 1. Слои и их параметры, используемые в модели CRNN
Table 1. Layers and their parameters used in the CRNN model

Слои	**Параметры**
Input	size = (64, 256, 1)
Convolutional	64 kernels, size = (3,3)
MaxPooling	Window – (2,2), strides = (2,2)
Convolutional	256 kernels, size = (3,3)
Convolutional	256 kernels, size = (3,3)
MaxPooling	Window – (2,1), strides = (2,1)
Convolutional	512 kernels, size = (3,3)
Batch_norm	–
MaxPooling	Window – (2,1), strides = (2,1)
Lambda	squeeze
B-LSTM cell	128
B-LSTM cell	128
Dropout	0.5
Dense	128
Dense	64
Dropout	0.5
Softmax	Количество символов
CTC	–

3.2 Алгоритм DA-CRNN

Собирать большое количество реальных изображений и размечать вручную для обучения модели CRNN – очень дорогой и долгий процесс.

Заметим, что нашу задачу можно переформулировать как задачу адаптации к предметной области: мы хотим, имея модель, показывающую хорошие результаты на искусственно сгенерированных данных, адаптировать ее для распознавания реальных изображений, используя только картинки этих реальных изображений и не зная ответов. Поэтому логично исследовать применимость методов адаптации к предметной области к нашей задаче.

Ярослав Ганин и соавторы в 2016 году опубликовали статью [9], в которой описывают метод адаптации к предметной области с помощью доменно-состязательной нейронной сети.

Для того, чтобы классификатор мог предсказывать реальные изображения также хорошо, как и искусственные, необходимо, чтобы внутреннее представление сети не имело никакой дискриминационной информации о происхождении картинки (реальная или искусственная картинка) при этом сохраняя низкую ошибку классификации на размеченных искусственных примерах.

Суть метода представляет из себя добавление к классификатору дискриминатора, который будет различать реальные картинки от искусственных. Архитектура состоит из 3 частей – часть отбора признаков, классификатор, дискриминатор; пусть $\theta_f, \theta_y, \theta_d$, соответственно, весовые коэффициенты этих частей. L_y, L_d – ошибки классификатора и дискриминатора соответственно. Общая ошибка задается взвешенной суммой этих ошибок, она минимизируется.

Идея состоит в том, чтобы дискриминатор работал таким образом, что принимая на вход результат части отбора признаков, он не мог различить реальную картинку от синтетической, то есть полученные признаки были свойственны как реальным так и не реальным изображениям. Грубо говоря, реализуется правильный и нужный отбор общих признаков, не заточенный под особенности искусственной картинки. Этого можно добиться, если обновлять коэффициенты θ_f, руководствуясь не уменьшением ошибки L_d (ход по антиградиенту), а руководствуясь увеличением ошибки L_d (ход по градиенту). То есть слой отбора признаков препятствует дискриминатору определять истинную природу картинки.

Но, в свою очередь, веса θ_d обновляются для уменьшения L_d. Тем самым поддерживается адекватный уровень работы дискриминатора после выхода признаков из первой части, но также поддерживается стремление удалить из выхода первой части какие-либо признаки, свойственные происхождению картинки (реальная или синтетическая).

Одновременно с этим классификатор обучается на синтетических картинках и дает определенные полезные признаки на выходе части отбора признаков.

Веса обновляются таким образом:

$$\theta_f \leftarrow \theta_f - \mu\left(\frac{\partial L_y}{\partial \theta_f} - \lambda\frac{\partial L_d}{\partial \theta_f}\right), \quad (2)$$

$$\theta_y \leftarrow \theta_y - \mu\frac{\partial L_y}{\partial \theta_y}, \quad (3)$$

$$\theta_y \leftarrow \theta_y - \mu\lambda\frac{\partial L_d}{\partial \theta_d}, \quad (4)$$

μ – скорость обучения, λ – вес ошибки дискриминатора.

$L_{common} = L_y + \lambda L_d$ – это общая функция потерь, которую мы минимизируем.

Оптимизация функции потерь происходила с помощью метода стохастического градиентного спуска. Параметр скорости обучения μ изменялся на каждой эпохе в соответствии с формулой (1).

Гиперпараметр веса ошибки дискриминатора λ фиксируется в пределах 5-20 и не изменяется в процессе обучения. Точное значение параметра λ мы подбираем с помощью валидации, то есть обучаем несколько моделей с разными значениями λ и выбираем ту, которая показывает наивысшее качество на валидационной выборке.

Для реализации метода, используется дополнительный слой – слой обращения градиента. Этот слой умножает на отрицательное число градиент функции потерь дискриминатора по θ_f при обратном распространении ошибки, но не изменяет веса при прямом проходе. Это и обеспечивает ход по градиенту, а не по антиградиенту для весов слоя отбора признаков θ_f. Математически этот слой описывается так:

$$R(x) = x, \frac{dR(x)}{dx} = -\lambda_p I, \quad (6)$$

где I – единичная матрица, а λ_p изменяется от 0 до 1 в течении обучения по следующей формуле:

$$\lambda_p = \frac{2}{(1 + e^{-10 \cdot p})} - 1, \text{где } p = \frac{\text{номер эпохи}}{\text{общее количество эпох}}. \quad (7)$$

Эта стратегия постепенного увеличения числа λ_p, во-первых, позволяет классификатору быть менее чувствительным к шумным данным на ранних этапах обучения, а во-вторых, не дает дискриминатору обучиться быстрее классификатора, то есть сохраняется нужный баланс.

Архитектура GAN, о которой мы говорили в подразделе 2.3 многоступенчатая; чтобы этого избежать, мы попробовали интегрировать архитектуру CRNN как основную модель в задачу адаптации к домену DANN (в дальнейшем будем называть получившуюся архитектуру DA-CRNN). Схема новой архитектуры приведена на рис. 1.

Параметры начальных сверточных слоев и верхней части новой модели DA-CRNN приведены в табл. 1, а отдельно параметры архитектуры дискриминатора приведены в табл. 2.

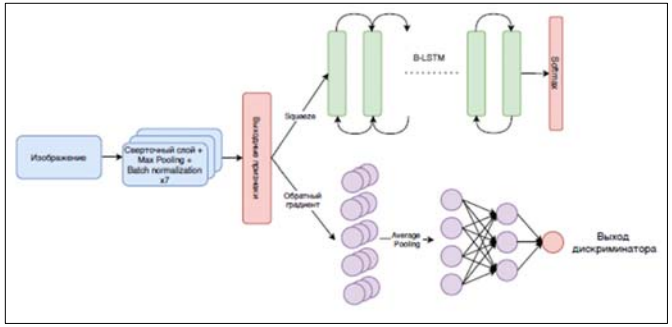

Рис. 1. Архитектура DA-CRNN
Fig. 1. DA-CRNN Architecture

Параметры начальных сверточных слоев и верхней части новой модели DA-CRNN приведены в табл. 1, а отдельно параметры архитектуры дискриминатора приведены в таблице табл. 2.

Табл. 2. Слои и их параметры, используемые в архитектуре дискриминатора
Table 2. Layers and their parameters used in the architecture of the discriminator

Слои	**Параметры**
Input	size = (32, 512)
Dense	64
Average Pooling	32
Lambda	squeeze
Dropout	0.5
Dense	16
Softmax	Ответ дискриминатора

4. Эксперименты

4.1 Набор данных

Датасет для обучения модели представляет из себя набор искусственно-сгенерированных размеченных капч в количестве 19 тысяч штук, а также набор реальных неразмеченных капч, собранных с сайтов Яндекс, Wikipedia и eBay, каждый набор в количестве 9000 штук, с помощью сборщика, написанного на языке Python с использованием библиотеки для автоматизации работы браузера Selenium WebDriver[1].

Длина текста на каждом наборе реальных данных разная, но всегда варьируется в районе 5-10 символов. Под символами могут пониматься английские и русские буквы, а также арабские цифры.

Примеры капч из Wikipedia, Яндекса, eBay изображены на рис. 2.

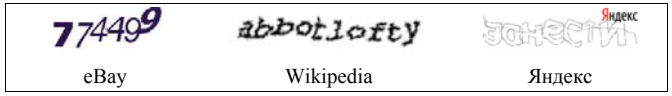

Рис. 2. Примеры реальных капч
Fig. 2. Examples of real captchas

[1] https://www.selenium.dev/

1500 тысячи капч из каждого набора реальных примеров были размечены вручную для обучения и тестирования модели. Для обучения использовалось 100-500 примеров, для тестирования всегда 1 тысяча.

Для генерации искусственных размеченных изображений в количестве 20 тысяч используется библиотека для генерации размеченных картинок Captcha[2]. Для каждого обучения на определенном наборе реальных данных использовались по возможности похожие по шрифту искусственно-сгенерированные размеченные изображения. Примеры сгенерированных капч изображены на рис. 3.

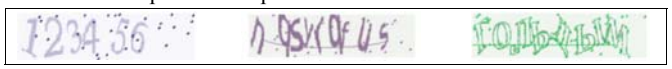

Рис. 3. Примеры синтетических капч
Fig,. 3. Examples of synthetic captchas

Валидационная выборка, использующаяся для тонкой настройки весов в методах CRNN и DA-CRNN, применяется для определения некоторых оптимальных гиперпараметров и для сохранения наилучшей модели с этими параметрами. Для модели CRNN это количество эпох при пост-обучении, а для DA-CRNN гиперпараметрами являются: количество эпох при непосредственном обучении, параметр λ – вес ошибки дискриминатора (см.подраздел 3.2), а также количество эпох пост-обучения.

4.2 Результаты модели CRNN на искусственных изображениях

Цель эксперимента состоит в том, чтобы выяснить применимость CRNN для разгадывания капчи, поэтому этот эксперимент проводился на синтетических капчах для упрощения получения размеченных примеров.

Также при обучении на синтетических данных была получена предобученная модель с полезными признаками для дальнейшего переноса знаний.

Для обучения моделей мы используем 19 тысяч сгенерированных размеченных шумных картинок, используя библиотеку для генерации капч Captcha (подробнее см. подраздел 4.1), то есть используем обучение с учителем. Далее модель тестируется на одной тысяче также искусственно сгенерированных изображений.

Результаты теста на искусственных сгенерированных данных после обучения модели приведено в табл. 3. Обучение и подсчеты производились с помощью видеокарты GeForce GTX 1080.

В роли метрики качества мы используем метрику «Достоверность», определенную как:

$$\text{Достоверность} = \frac{\text{множество правильно отгаданных тестовых изображений}}{\text{множество всех тестовых изображений}}. \quad (8)$$

Табл. 3. Результаты модели CRNN на синтетических данных
Table 3. Results of the CRNN model on synthetic data

	Время обучения, м	**Достоверность, %**	**Среднее время разгадывания 1 капчи, с**
CRNN	85	97	9,0016

Стоит отметить, что время отгадывания таких изображений человеком примерно 10-15 секунд, тем самым существует большой смысл использовать нейронные сети для решения задачи сбора информации из Интернета для уменьшения вносимой латентности.

[2] https://github.com/lepture/captcha

4.3 Результаты модели DA-CRNN на реальных данных

Проводится обучение новой построенной модели DA-CRNN на 19 тысячах сгенерированных капчах, при этом для адаптации мы используем 9 тысяч неразмеченных реальных картинок с сайтов. После этого, веса тонко настраиваются на 100-500 размеченных реальных изображениях. Полученная модель тестируется на 1000 реальных картинках (подробно о датасете можно прочитать в подразделе 4.1). Для чистоты эксперимента мы проводили по 5 независимых процессов обучения, в каждом из которых случайно выбирался поднабор тренировочных, валидационных и тестовых выборок. Валидационный набор считался используемым в обучении, то есть если используется n размеченных примеров для тонкой настройки, то 0.2\% от них используется для валидации (более подробно про валидацию см. в разделе 4.1). Результаты тестирования качества распознавания приведены на рис. 4.

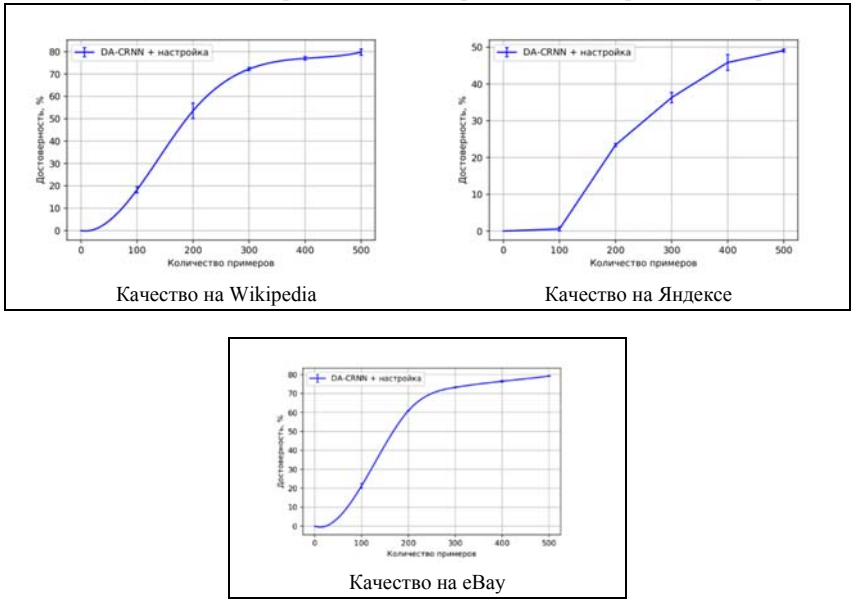

Рис. 4. Результаты модели DA-CRNN на капчах из Wikipedia, Яндекс и eBay
Fig. 4. Results of the DA-CRNN model on captchas from Wikipedia, Yandex and eBay

Заметим, что для достижения точности в 50% для капч из Wikipedia и eBay требуется всего 200 размеченных реальных примеров для обучения. А качество в 50\% показывают многие сторонние методы из литературы только на 500 размеченных реальных данных.

4.4 Сравнение с современными методами

Сравним теперь результаты распознавания нашей модели и самых эффективных на данный момент методов. Показатели достоверности, скорости работы и обучения взяты из соответствующих статей.

В методах [7-8] модели DA-CRNN тонко настраивались на 500 размеченных изображениях и тестировались на 1000 размеченных изображениях. Сравнение методов представлено в таблице 4. Стоит также отметить, что один из самых эффективных на данный момент методов на основе GAN [7] работает лишь с изображениями фиксированной длины, поэтому для теста на представленных сайтах авторами брался поднабор с фиксированной длиной, т.е. задача упрощалась. А вот метод [8] предполагает распознавание капч с произвольной длиной.

Табл. 4. Сравнение качества модели DA-CRNN с современными методами
Table 4. Comparison of the quality of the DA-RNN model with modern methods

Метод	Wikipedia	Яндекс	eBay
[22]	23.8%	–	58.8%
[23]	25%	–	43%
[3]	28%	–	51.39%
[24]	–	56.0%	–
[8]	66.5%	63.2%	**91.5%**
[7]	78%	–	85.6%
DA-CRNN	**79%**	**72.26%**	79.3%

Как мы видим, наша модель DA-CRNN превосходит множество ранее известных методов на некоторых наборах реальных данных и имеет качество, сравнимое с наилучшими на данный момент на остальных. Реализация метода DA-CRNN не требует априорного знания длины капчи и дополнительную предобработку изображения. Процесс обучения использует одну функцию потерь и имеет одноступенчатую архитектуру, тем самым снижается время обучения. В табл. 5 представлено сравнение времени обучения и тестирования методов, показавших наибольшую достоверность.

Табл. 5. Время обучения и время тестирования современных методов
Tab. 5. Training time and testing time of modern methods

Метод	Ресурсы	Время на обучение, ч	Время разгадывания 1 капчи, мс
[7]	4 x NVIDIA Tesla P40 GPU, 256GB of RAM	53	50
[8]	2 x NVIDIA Tesla P40 GPU	–	6
DA-CRNN	NVIDIA GeForce GTX 1080, 64GB of RAM	**12**[3]	**1,6**

Из табл. 5 видно, что наш метод DA-CRNN, имея одну видеокарту, затрачивает на обучение и тестирование гораздо меньше времени, чем остальные методы с более сильными ресурсами. Это происходит благодаря простоте модели и относительно небольшому числу примеров для обучения. Например, в статье [7]} используется 200000 сгенерированных картинок, а в методе [8] используется 800000, в нашем же методе всего 19000. Исходя из этого, можно предположить, что метод [8] обучается на порядок дольше метода DA-CRNN.

Беря во внимание быстроту нашей модели, можно объяснить более низкий процент качества DA-CRNN на наборе капч из сайта eBay. Модель показывает сравнимое качество распознавания на всех рассматриваемых наборах данных, при этом затрачивая наименьшее время на обучение и распознавание.

В итоге, мы можем наблюдать, что для достижения точности в 50\% (что дает нам математическое ожидание числа попыток до успешного обхода капчи, равное двум) нам требуется всего 200-500 размеченных реальных примеров, что делает нашу модель релевантной и практичной.

5. Заключение

В данной работе был представлен метод распознавания текста на изображениях, содержащих текстовую капчу с текстом произвольной длины. Было экспериментально показано, что метод, основанный на архитектуре нейронной сети CRNN и использующий подход к

[3] 12 часов требуется для полного цикла обучения подбором всех оптимальных гиперпараметров. Для быстрого получения не самой точной модели за одну итерацию обучения потребуется около 30 минут.

адаптации к предметной области DANN достигает качества распознавания, не уступающее другим эффективным методам.

Главные преимущества построенного метода в том, что он не требует априорного знания длины капчи и использует для обучения 200-500 реальных размеченных примеров. Также к достоинствам данного метода можно отнести то, что он состоит из одной нейросетевой модели и не требует серьезной предобработки изображения. Процесс настройки модели включает в себя два этапа -- обучения начального приближения и тонкой настройки на небольшом количестве реальных примеров, что позволяет уменьшить время обучения примерно в 5 раз по сравнению с современными методами.

Мы добились высокого качества распознавания новой модели DA-CRNN на нескольких наборах реальных изображений с разных сайтов. В будущем хотелось бы улучшить это качество, использовать еще меньше примеров для обучения для снижения времени обучения, а также провести дополнительный ряд экспериментов с наборами данных других поставщиков реальных капч с веб-ресурсов, таких как Вконтакте, Qihu360, cadocs.nsd.ru, Alipay, Baidu, Sina, Google. каждой строки можно определять её уровень вложенности по отношению к документу.

Список литературы / References

[1] А.К. Яцков, М.И. Варламов, Д.Ю. Турдаков. Сбор и извлечение данных с веб-сайтов СМИ. Программирование, том 44, №5, 2018 г., стр. 68-80 / A. K. Yatskov, M. I. Varlamov и D. Y. Turdakov. Extraction of data from mass media web sites. Programming and Computer Software, vol. 44, № 5, 2018, pp. 344–352.

[2] M. Kopp, M. Nikl, and M. Holena. Breaking captchas with convolutional neural networks. In Proc. of the 17th Conference on Information Technologies – Applications and Theory (ITAT 2017), 2017, pp. 93-99.

[3] E. Bursztein, J. Aigrain, and A. Moscicki. The end is nigh: generic solving of text-based captchas. In Proc. of the 8th USENIX Workshop on Offensive Technologies, 2014, 15 p.

[4] T. A. Le, A. G. Baydin, R. Zinkov и F. D. Wood. Using synthetic data to train neural networks is model-based reasoning. arXiv: 1703.00868, 2017.

[5] X. Wang, M. You, and C. Shen. Adversarial generation of training examples for vehicle license plate recognition. arXiv: 1707.03124, 2017.

[6] F. Zhan, C. Xue, and S. Lu. GA-DAN: geometry-aware domain adaptation network for scene text detection and recognition. arXiv: 1907.09653, 2019.

[7] G. Ye, Z. Tang, D. Fang, Z. Zhu, Y. Feng, P. Xu, X. Chen, and Z. Wang. Yet another text captcha solver: A generative adversarial network based approach. In Proc. of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 332-348

[8] S. Tian and T. Xiong. A generic solver combining unsupervised learning and representation learning for breaking text-based captchas. In Proc. of the Web Conference, 2020, pp. 860-871.

[9] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V.S. Lempitsky. Domain-adversarial training of neural networks. Journal of Machine Learning Research, volo. 17, 2016, pp. 1-35.

[10] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556, 2014.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv: 1512.03385, 2015.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, vol. 86, no. 11, 1998, pp. 2278—2324.

[13] Botdetect captcha generator [online]. URL: www.captcha.com, accessed 25.08.2020.

[14] F. Stark, C. Hazırba， R. Triebel, and D. Cremers. Captcha recognition with active deep learning. In Proceedings of the German Conference on Pattern Recognition, Workshop on new challenges in neural computatio, 2015.

[15] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial networks. ArXiv: 1406.2661, 2014.

[16] A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. arXiv: 1807.03748, 2018.

[17] B. Shi, X. Bai, and C. Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. arXiv: 1507.05717, 2015.
[18] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, vol. 9, no. 8, 1997, pp. 1735—1780.
[19] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In Proc. of the Twenty-Third International Conference on Machine Learning, 2006, pp. 369-376.
[20] T. Bluche, J. Louradour, and R. O. Messina. Scan, attend and read: end-to-end handwritten paragraph recognition with MDLSTM attention. arXiv: 1604.03286, 2016.
[21] S. Ruder. An overview of gradient descent optimization algorithms. arXiv: 1609.04747, 2016.
[22] H. Gao, J. Yan, F. Cao, Z. Zhang, L. Lei, M. Tang, P. Zhang, X. Zhou, X.Wang, and J. Li. A simple generic attack on text captchas. In Proc. of the Network and Distributed System Security Symposium, 2016, 14 p.
[23] E. Bursztein, M. Martin, and J. Mitchell. Text-based captcha strengths and weaknesses. In Proc. of the 18th ACM Conference on Computer and Communications Security, 2011, pp. 125-138.
[24] M. Tang, H. Gao, Y. Zhang, Y. Liu, P. Zhang, and P. Wang. Research on deep learning techniques in breaking text-based captchas and designing image-based captcha. IEEE Transactions on Information Forensics and Security, vol. 13, no.10, 2018, pp. 2522—2537.

Информация об авторах / Information about authors

Денис Олегович КУЩУК – студент магистратуры Физтех школы прикладной математики и информатики. Научные интересы: машинное обучение, состязательное обучение, оптическое распознавание символов.

Denis Olegovitch KUSHCHUK – Master's student at the Phystech School of Applied Mathematics and Informatics. Research Interests: Machine Learning, Adversarial Learning, Optical Character Recognition.

Максим Алексеевич РЫНДИН – аспирант. Научные интересы: методы адаптации к домену и переноса знаний, онлайн обучение, обработка текстов на естественном языке, векторное представление слов/предложений/текстов, генеративные модели, активное и проактивное обучение, анализ социальных сетей.

Maxim Alexeevitch RYNDIN – PhD Student. Research interests: domain adaptation, transfer learning, online learning, natural language processing, embeddings, generative models, active and proactive learning, social media analysis.

Александр Константинович ЯЦКОВ – аспирант. Научные интересы: сбор данных из веба, автоматизация процесса сбора данных, фокусированный сбор данных извлечение информации, машинное обучение.

Alexander Konstantinovitch YATSKOV – PhD Student. web crawling, web data extraction); focused crawling, information extraction, machine learning.

Максим Игоревич ВАРЛАМОВ – младший научный сотрудник. Научные интересы: автоматизация сбора данных из веб-ресурсов, автоматическое реферирование, семантическая близость понятий.

Maksim Igerevitch VARLAMOV – junior researcher. Research interests: automation of data collection from web resources, automatic summarization, semantic proximity of concepts.