ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ ПРИ РАСПОЗНАВАНИИ РУКОПИСНЫХ СИМВОЛОВ

HANDWRITTEN CHARACTER RECOGNITION USING NEURAL NETWORK

УДК: 004.93

Доронин Юрий Дмитриевич, магистрант, Московский Государственный Технический Университет имени Н.Э. Баумана, г. Москва

Doronin Yuri Dmitrievich, Graduate Student, Bauman Moscow State Technical University, Moscow. E-mail: doronin.yurii.d@gmail.com

Аннотация

Современные методы машинного обучения в совокупности с технологиями компьютерного зрения и OCR позволяют достичь высокого уровня качества при оцифровке документов на бумажных носителях. Одним из ограничивающихфакторов является объем тренировочных данных, который для некоторых языков может быть недостаточен для обучения CNN. Так, появилась необходимость в реализации модели с высокой точностью распознавания, не зависящей от большого объема набора данных.

Annotation

Modern methods of machine learning, combined with computer vision and OCR technologies, allow you to achieve a high level of quality when digitizing documents on paper. One limiting factor is the amount of training data, which for some languages may not be sufficient for CNN training. Therefore, it became necessary to implement a model with high recognition accuracy that does not depend on a large amount of data set.

Ключевые слова: нейронные сети, наборы данных, машинное обучение, рукописный текст

Key words: neural networks, data sets, machine learning, handwritten text

Modern methods of machine learning, combined with computer vision and OCR technologies, allow you to achieve a high level of quality when digitizing documents on paper. One limiting factor is the amount of training data, which for some languages may not be sufficient for CNN training. Therefore, it became necessary to implement a model with high recognition accuracy that does not depend on a large amount of data set.

Введение

Благодаря последним достижениям в области моделей глубокого обучения распознавание рукописных символов является практически решенной проблемой для многих наиболее используемых языков. Тем не менее, из-за отсутствия достаточно больших наборов данных, которые необходимы для обучения моделей глубокого обучения, для многих других языков распознавание рукописных цифр остается трудно решаемой задачей. Хотя для ее решения можно использовать общепринятые модели: линейные и нелинейные классификаторы, метод K-ближайших соседей и метод опорных векторов (SVM), они не в состоянии достичь производительности близкой к человеческому уровню, обеспечиваемой моделями глубокого обучения. Сверточные нейронные сети (CNN) достигли очень высокого уровня результативности благодаря своей способности кодировать глубокие признаки и пространственное понимание. Несмотря на то, что CNN хорошо распознают низкоуровневые и высокоуровневые функции в изображениях, они теряют ценную информацию на уровнях объединения слоев. CNN требуют большого количества обучающих выборок, обычно в масштабе тысяч или десятков тысяч на класс для успешного обучения и классификации изображений. В результате наблюдается большая заинтересованность в обучении CNN меньшим

количеством обучающих выборок.

В рамках научно-исследовательской работы рассмотрен метод, который решает проблему небольшого размера помеченного набора данных с помощью капсульных сетей.

1 Распознавание символов с помощью капсульной сети

Для решения задачи распознавания символов используется архитектура, состоящая из капсульной сети и сети декодера, как показано на рисунках 1 и 2.

В капсульной сети первые три уровня представляют собой сверточные уровни с 64 ядрами 3×3 с шагом 1 пиксель, 128 ядер 3×3 с шагом 1 и 256 ядер 3×3 с шагом 2 соответственно. Четвертый слой представляет собой первичный слой капсулы с тридцатью двумя каналами 8-мерных капсул, каждая первичная капсула содержит 8 сверточных единиц с ядрами 9×9 и шагом 2. Пятый слой, называемый слоем капсулы символа, представляет собой полностью связанный капсульный слой с 16-мерной капсулой для каждого класса, в результате получается M капсул для набора данных с M количеством классов.

В рассмотренной технологии используется динамическая маршрутизация между первичным слоем капсулы и символьным слоем капсулы, как предложено в [4], с тремя итерациями маршрутизации. На вход в капсульную сеть подается набор J, 28×28 изображений, выход - J×M×16 размерный тензор C, содержащий соответствующие параметры создания экземпляров, где каждый Cj, j ∈ [J] является матрицей параметров экземпляра j обучающей выборки. Перед передачей C в качестве входных данных в сеть декодера соответствующие параметры создания должны быть замаскированы нулями для всех классов, кроме истинного класса. Следовательно, замаскированный тензор C по-прежнему является J×M×16 размерной матрицей, но включающей только параметры создания экземпляров, соответствующие истинному классу, как ненулевые значения.

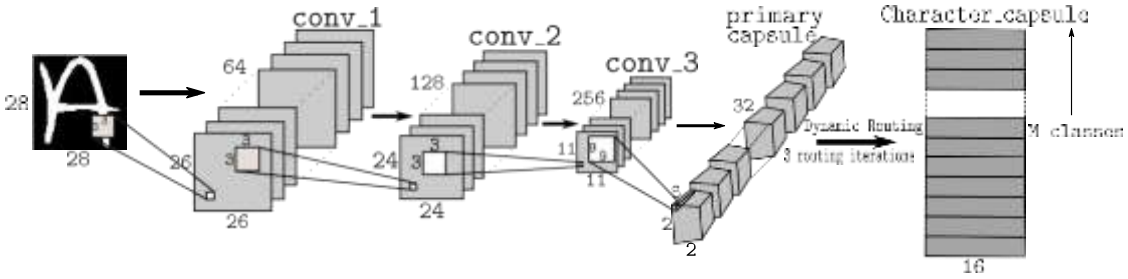

Рисунок 1. Предполагаемая модель TextCap для распознавания символов.

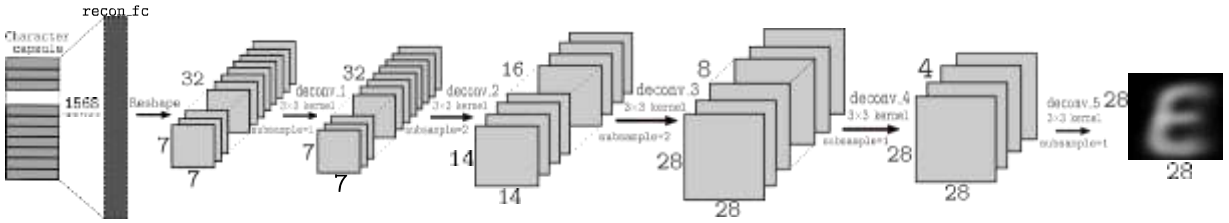

Рисунок 2. TextCap декодер. Связи декодеров для реконструкции символов.

Сеть декодера состоит из одного полностью связанного слоя, за которым следуют пять транспонированных сверточных слоев с параметрами, показанными нарис. 2. Входом в декодер является замаскированная матрица C, а выходом декодера - набор реконструированных 28 × 28 изображений. За исключением последнего слоя деконволюции, который имеет сигмовидную активацию, полностью связанный слой и другие слои деконволюции имеют активацию ReLU.

Сначала обучается предложенная модель на полных обучающих наборах и оценивается ее производительность. Впоследствии, пытаясь решить проблему отсутствия большого количества обучающих выборок для распознавания символов и аналогичных задач, описанных в разделе 1, достигается высокий уровень производительности, используя 200 обучающих образцов на класс, используя ту же сеть.

Изучая результаты вышеприведенного раздела с небольшим количеством обучающих выборок, даже несмотря на то, что производительность капсульной сети достигла очень высокого уровня, декодирующая сеть не может достичь приемлемой реконструкции. Наиболее очевидным решением для повышения производительности сети декодера является увеличение количества обучающих выборок путем создания новых из уже имеющихся в исходном наборе

обучения. В связи с этим используется метод создания новых выборок путем дополнения с помощью концепции параметров создания экземпляров в CapsNets, который описан в статье [1].

2 Результаты распознавания

Для каждого набора данных в таблице 1 TextCaps обучается на 200 обучающих выборках из набора и тестируется, используя весь набор тестов. Чтобы проверить производительность архитектуры TextCaps, необходимо оценить модель, обучая ее на полных обучающих наборах и тестируя на полных наборах тестов.

Таблица 1. Пять наборов данных для обучения TextCaps

Набор данных	Кол-во классов	Кол-во тренировочных классов	Размер тренировочного набора	Размер теста
EMNIST-Balanced[7]	47	2,400	112,800	18,800
EMNIST-Letters[7]	26	4,800	124,800	20,800
EMNIST-Digits[7]	10	24,000	240,000	40,000
MNIST[5]	10	6,000	60,000	10,000
Fashion MNIST[22]	10	6,000	60,000	10,000

2.1 Классификация рукописных символов

В таблице 2 результаты сравниваются с наилучшими достигнутыми. В неё включены результаты, полученные с использованием полных обучающих наборов, атакже с использованием только 200 обучающих выборок на класс. В обоих случаях использованы полные наборы тестов, показанные в таблице 1, чтобы указать среднюю точность. Используется комбинация предельной потери и потери реконструкции для обучения, как предложено в [2], и, более того, процедура обучения, выполняемая для каждого эксперимента в этой статье аналогична [2]. Для каждого набора данных используется ансамблирование модели, чтобы повыситьточность и избежать чрезмерной подгонки. Используем циклические скорости обучения для каждых 30 эпох, что дает 3 ансамблевые модели с 90 эпохами [23].

Таблица 2. Сравнение TextCaps с иными современными моделями, показаны среднеезначение и стандартное отклонение из 3 испытаний.

EMNIST-Letters		
Модель	Полный тестовый набор	200 наборов на класс
Cohen et al. [7] Wiyatnoet al.[9]	85.15% 91.27%	- -
TextCaps	95.36 ± 0.30%	92.79 ± 0.30%
EMNIST-Balanced		
Модель	Полный тестовый набор	200 наборов на класс
Cohen et al. [7] Dufourq et al. [8]	78.02% 88.3%	- -
TextCaps	90.46 ± 0.22%	87.82 ± 0.25%
EMNIST-Digits		
Модель	Полный тестовый набор	200 наборов на класс
Cohen [7] Dufourq [8]	95.90% 99.3%	- -
TextCaps	99.79 ± 0.11%	98.96 ± 0.22%
MNIST		
Модель	Полный тестовый набор	200 наборов на класс
Sabour [2] Cires [6] Wan	99.75% 99.77% 99.79%	- - -
TextCaps	99.71 ± 0.18%	98.68 ± 0.30%
Fashion MNIST		
Модель	Полный тестовый набор	200 наборов на класс
Xiao Bhatnagar Zhong	89.7% 92.54% 96.35%	- - -
TextCaps	93.71 ± 0.64%	85.36 ± 0.79%

Сначала описываются результаты, полученные с помощью полных обучающих наборов, и сравниваются с наилучшими достигнутыми. По EMNIST-letters рассмотренный метод опережает Wiyatno на 4,09%. Система для

сбалансированного набора данных EMNIST достигла средней точности 90,46%, что превосходит наилучшую Dufourq на 2,16%. Для набора данных EMNIST-digits TextCaps рассмотренной системе удалось превзойти достижения Dufourq на 0,49%. Для MNIST и Fashion-MNIST система обеспечила высочайшую точность. Тем не менее, результаты находятся на одном уровне.

Затем сравниваются результаты, полученные только с 200 обучающими выборками на класс. Для EMNIST-letters система превосходит наилучшие имеющиеся результаты на 1,52%. Тем не менее для EMNIST-balanced, EMNIST-digits,MNIST удалось достичь лучших результатов.

2.2 Результаты тестирования функции потерь

Далее мы обсудим варианты реконструкции с учетом различных функций потерь и комбинаций функций потерь. Все рассматриваемые модификации протестированы с различным количеством обучающих выборок (100, 200, 500 и 1000) для каждого класса из набора данных EMNIST-balanced. Для этого анализа использована модель CapsNet, предложенную в [4] с небольшими изменениями, где сеть декодера состоит из полностью связанных слоев, поскольку эта архитектура хорошо известна.

Рисунок 3 иллюстрирует изменение PSNR в зависимости от количества обучающих выборок, используемых для различных функций потерь, что приводит к ряду интересных наблюдений. Например, когда количество обучающих выборокневелико (100 или 200), производительность L1 низкая по сравнению с MSE, но для большего количества обучающих выборок L1 работает лучше. Значения PSNR для BCE являются самыми высокими независимо от количества обучающих выборок. Следовательно, наиболее подходящей функцией потери для реконструкции потери вцелом является BCE.

Также, если линейно комбинировать две функции потерь для построения модифицированной функции потерь, полученные реконструкции будут относительноплохими. Поскольку разные функции потерь захватывают разные свойства рукописных символов, использованы два декодера с двумя функциями потерь и объединены два восстановленных изображения вместе, чтобы

получить результирующее восстановленное изображение с улучшенным качеством. Эта модификация показала, что отдельные реконструкции для соответствующих функцийпотерь также улучшились из-за влияния компонента потерь восстановления на потери при обучении. Однако степень улучшения индивидуальных PSNR зависит от комбинации функции потерь. Таблица 3 демонстрирует, как отдельные значения PSNR улучшаются с использованием сетей с двумя декодерами для различных комбинаций функций потерь восстановления.

Таблица 3. Значения PSNR для отдельных реконструкций при использовании различных комбинаций функций потерь.

Комбинации функций потерь	Количество тренировочных классов			
	100	200	500	1000
L_1 & DSSIM	13.51 12.89	14.64 14.19	15.95 15.57	17.48 17.03
L_1 & BCE	14.33 14.57	15.26 15.44	16.60 16.71	18.10 18.13
MSE & DSSIM	13.87 12.95	14.81 14.06	16.00 15.47	17.29 16.79
MSE & BCE	14.58 14.59	15.19 15.20	16.55 16.56	17.76 17.78
BCE & DSSIM	14.62 13.80	15.41 14.77	16.78 16.24	18.08 17.61

С двумя функциями потерь качество финальных реконструкций было намного лучше, и значения PSNR значительно улучшились по сравнению с моделью с одним декодером с одной функцией потерь. На рисунке 4 показано улучшение PSNR при финальном восстановлении вывода, когда две функции потерь объединяются вместе с помощью сети с двумя декодерами. На рисунке 14 показаны вариации восстановленных изображений с использованием различных комбинаций функций потерь для 200 обучающих выборок

.

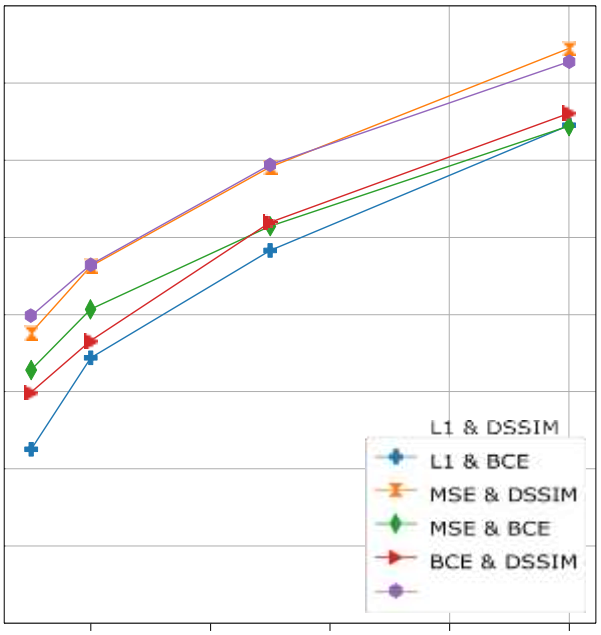

Рисунок 3 – Изменение PSNR для различных функций потерь реконструкции.

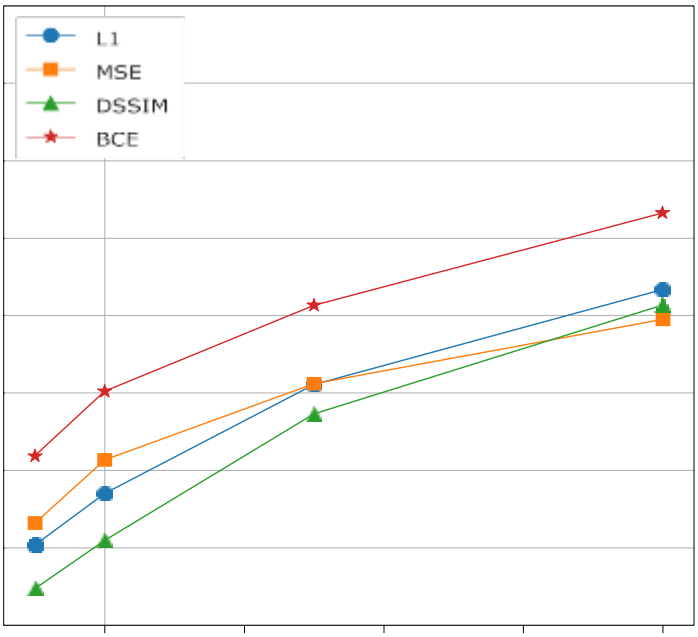

Рисунок 4 - Изменение PSNR для различных комбинаций функций потерь.

На рисунке 3 видно, что BCE показывает себя лучше по сравнению с другими функциями потерь при использовании и в модели сети с одним декодером и в модели с двумя. Рисунок 4 показывает, что комбинации BCE&DSSIM и L1&BCE работают значительно лучше, чем другие комбинации потерь для модели с двумя декодерами. Несмотря на то, что значения PSNR для потерь DSSIM не были достаточно значительными, оно отражает аспекты

пространственного сходства при реконструкции. Следовательно, комбинация потерь BCE & DSSIM обеспечивает незначительно лучшие реконструкции по сравнению с L1&BCE для меньшего количества обучающих выборок. Однако, когда количество обучающих выборок увеличивается, комбинация L1&BCE дает гораздо лучшие реконструкции.

3 Заключение

В этой статье исследована техника увеличения размера набора данных за счет использования концепций капсульных сетей. Продемонстрирована эффективность этой техники на хорошо изученных наборах рукописных символьных данных.

Литература

1. Информационная база Researchgate [Электронный ресурс]. - 2017. - URL: https://www.researchgate.net/publication/267960550_ImageNet_Classification_with_Dee p_Convolutional_Neural_Networks (дата обращения 03.06.2020);
2. Информационная база psycnet [Электронный ресурс]. - 2017. - URL: https://psycnet.apa.org/record/1992-11189-001 (дата обращения 26.05.2020);
3. Информационная база института Cornell [Электронный ресурс]. - 2017. - URL: https://arxiv.org/abs/1710.09829 (дата обращения 03.06.2020);
4. Информационная база института Cornell [Электронный ресурс]. - 2017. - URL: https://arxiv.org/abs/1710.09829 (дата обращения 03.06.2020);
5. Информационная база MNIST [Электронный ресурс]. - 2017. - URL: http://yann.lecun.com/exdb/mnist/ (дата обращения 03.06.2020);
6. Информационная база института Cornell [Электронный ресурс]. - 2017. - URL: https://arxiv.org/abs/1202.2745 (дата обращения 03.06.2020);
7. Информационная база института Cornell [Электронный ресурс]. - 2017. - URL: https://arxiv.org/abs/1702.05373 (дата обращения 03.06.2020);
8. Информационная база института Cornell [Электронный ресурс]. - 2017. - URL: https://arxiv.org/abs/1709.09161 (дата обращения 03.06.2020);
9. Информационная база института Cornell [Электронный ресурс]. - 2017. - URL: https://arxiv.org/abs/1803.01900 (дата обращения 03.06.2020)

Literature

1. Researchgate information base [Electronic resource]. - 2017. - URL: https://www.researchgate.net/publication/267960550_ImageNet_Classification_with_Dee p_Convolutional_Neural_Networks (date of access 03.06.2020);
2. Information base psycnet [Electronic resource]. - 2017. - URL: https://psycnet.apa.org/record/1992-11189-001 (date of access 05/26/2020);
3. Information base of the Cornell Institute [Electronic resource]. - 2017. - URL: https://arxiv.org/abs/1710.09829 (date of treatment 06/03/2020);
4. Information base of the Cornell Institute [Electronic resource]. - 2017. - URL: https://arxiv.org/abs/1710.09829 (date of treatment 06/03/2020);
5. Information base MNIST [Electronic resource]. - 2017. - URL: http://yann.lecun.com/exdb/mnist/ (date of treatment 06/03/2020);
6. Information base of the Cornell Institute [Electronic resource]. - 2017. - URL: https://arxiv.org/abs/1202.2745 (date of treatment 06/03/2020);
7. Information base of the Cornell Institute [Electronic resource]. - 2017. - URL: https://arxiv.org/abs/1702.05373 (date of treatment 06/03/2020);
8. Information base of the Cornell Institute [Electronic resource]. - 2017. - URL: https://arxiv.org/abs/1709.09161 (date of treatment 06/03/2020);
9. Information base of the Cornell Institute [Electronic resource]. - 2017. - URL: https://arxiv.org/abs/1803.01900 (date of access 03.06.2020)