

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
Факультет Бизнеса и Менеджмента

**РАСПОЗНАВАНИЕ СИМВОЛОВ НА КЛИНОПИСНЫХ ТАБЛИЦАХ ПРИ
ПОМОЩИ МОДЕЛЕЙ ГЛУБИННОГО ОБУЧЕНИЯ**

Исследовательская работа студентов
3 курса, бакалавриата образовательной программы
«Бизнес-информатика»
Алвандяна Нарека Аршаковича
Кана Максима Валериевича
Сагайдачного Леонида Юрьевича

Научный руководитель
Определенов В. В.

Москва 2020

Содержание

1. Введение
2. Теория
 - 2.1. Задача детекции
 - 2.2. Сверточные нейронные сети
3. Обзор основных работ в области детекции
4. Подходы к решению задачи
 - 4.1. Задача распознавания символов как задача детекции
 - 4.2. Задача распознавания символов на основе сравнения графов
 - 4.3. Задача распознавания символов как задача обработки естественного языка
5. Направления будущих исследований
6. Заключение
7. Список литературы

1. Введение

Клинопись на данный момент считается наиболее ранней из всех известных систем письма. Данная письменность была создана около 3000 года до н.э. На глиняных табличках, пока глина еще не затвердела, выводились знаки с помощью палочек для письма или же любого заостренного приспособления, отчего и появлялись «клинообразные» штрихи. Изначально писали справа налево, колонками, затем слева направо строками.

Впервые клинописные таблички начали находить в Европе 17 века, но у ученых не было возможности их прочесть или перевести. Попытки дешифровать клинопись были сделаны датчанином Карстеном Нибуrom в конце 18 века. Однако Нибуру не удалось выполнить эту задачу. Он мог лишь установить, что клинообразные надписи писались при помощи трёх систем письменности и что наиболее простая из них состоит из 42 алфавитных знаков.

Прорывом в данной области можно считать работы Георга Фридриха Гротефенда, который смог проанализировать и доказать, что клинопись является именно письменностью, а не каким-то орнаментом. Также он сумел правильно разобрать девять алфавитных знаков древнеперсидской клинописи и тем самым положил начало дешифровке клинообразных надписей. Позже правильность его выводов была подтверждена другими учеными.

Нынешняя ситуация такова, что около 90 процентов табличек так и остаются непереуведенными. Основной причиной можно считать нехватку профессионалов в данной области, переводить таблички может лишь небольшая группа ученых. Решением этой проблемы может стать автоматический машинный переводчик.

На данный момент существуют относительно успешные алгоритмы, например изучающий культуру Шумера археолог-лингвист Эмили Паже-Перрон координирует проект по машинному переводу 69 000 месопотамских административных документов, старейший из которых датируется 21-м веком до нашей эры. Цель заключается не только в расшифровке текстов и предоставления их содержания мировой общественности. Паже-Перрон считает, что на основе клинописи можно будет в дальнейшем расшифровать и другие мёртвые языки. На примере 4 тысяч административных текстов из оцифрованной базы данных Паже-Перрон и ее команда обучают машинный переводчик.

Нельзя не упомянуть важность проектов в области оцифровки, так как без них машинный перевод был бы просто невозможен. Новые технологии облегчают работу с древними и часто поврежденными источниками. Канадские ученые оцифровывают таблички из музейных и частных коллекций

и создают «Цифровую библиотеку клинописных текстов». Обширная онлайн-база уже содержит около трети всех известных источников такого рода.

2. Теория

2.1. Задача детекции

Детекция – задача крайне похожая на локализацию, за исключением того, что на одном изображении одновременно может находиться более одного объекта, и каждый нужно правильно локализовать и классифицировать.

2.2. Сверточные нейронные сети

Сверточные сети – вид нейронных сетей, основу работы которого составила схема, по которой работает зрение у многих животных, и человека - в частности. Наше зрение так же, как и нейросеть состоит из нескольких слоев. После слоя, которые воспринимают информацию максимально детально идет слой, который оставляет из этой информации только самое важное и передает это следующему слою и так они чередуются 2-3 раза. Похожую схему перенесли и на искусственные нейронные сети.

Каждое изображение представляет из себя тензор размера $W \times H \times 3$, где W – количество пикселей в ширину H – количество пикселей в высоту, 3 – количество каналов (если речь идет об RGB изображении).

Входное изображение обрабатывается операцией свертки. Сворачиваются изображения отдельно по каждому каналу. Соответственно работает операция свертки с матрицами. Для того, чтобы свернуть изображение, необходимо ядро свертки (или фильтр), которое представлено матрицей.

Сама операция представляет собой поэлементное умножение соответствующих значений в матрице входного изображения на матрицу фильтра, суммирования полученных значений и запись их в новую матрицу, которая называется feature map. Ядро свертки проходит так по всему изображению скользящим окном с определенной шириной шага (stride). Процесс свертки фильтром 3×3 показан на рисунке 1. После свертки, все значения в новой feature map поэлементно проходят через какую-то нелинейную функцию активации.

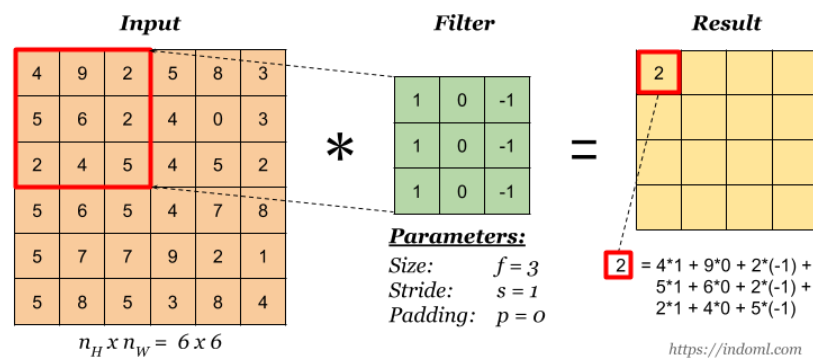


Рисунок 1. Пример работы операции свертки с фильтром 3x3. [Источник](#)

Сверточный слой имитирует ту часть человеческого зрения, которая отвечает за запоминание максимального числа деталей, для фильтрации важной информации после сверток добавляются Pooling слои. Этот слой так же скользящим окном проходит по feature map и в зависимости от размера окна описывает все числа, попавшие в окно, одним числом. Это может быть максимум из всех чисел (Max Pooling), среднее всех чисел (Average Pooling) или любая другая функция, имеющая под собой логичное обоснование. На рисунке 2 показана операция Max Pooling с окном 2x2 и шагом 2.

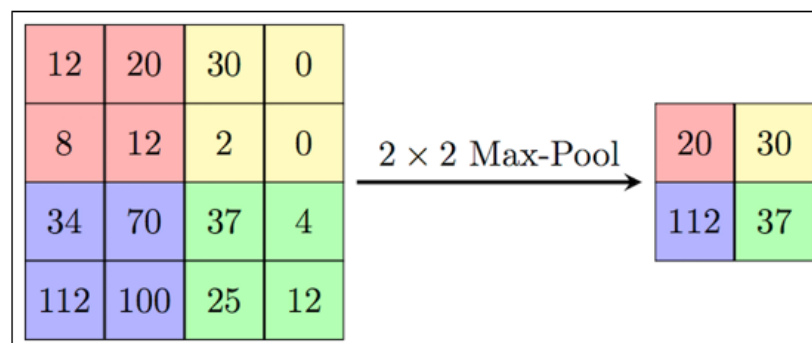


Рисунок 2. Пример работы операции свертки с фильтром 3x3. [Источник](#)

Сверточные сети в целом состоят из какого-то количества сверточных и pooling слоев, после которых финальную feature map матрицу чаще всего разворачивают в одномерный вектор и подают в качестве входа в полносвязные слои, которые и выполняют классификацию/регрессию.

В процессе обучения градиентным спуском обновляются значения в фильтрах свертки.

Главное преимущество сверточных сетей заключается в том, что в отличие от полносвязных слоев, им не важно местонахождения объекта, то есть если во время обучения интересующий объект находился во всех локациях, кроме нижнего левого угла, то во время тестирования сверточная сеть все равно (скорее всего) узнает там нужный объект и присвоит ему правильный класс.

3. Обзор классических работ в области детекции

3.1.R-CNN (Regions with CNN)

Данная модель [8] является первой глубокой нейронной сетью для решения задачи детекции ощутимо превзошедшей последнее на тот момент state-of-the-art решение. Схема работы алгоритма показана на рисунке 3.

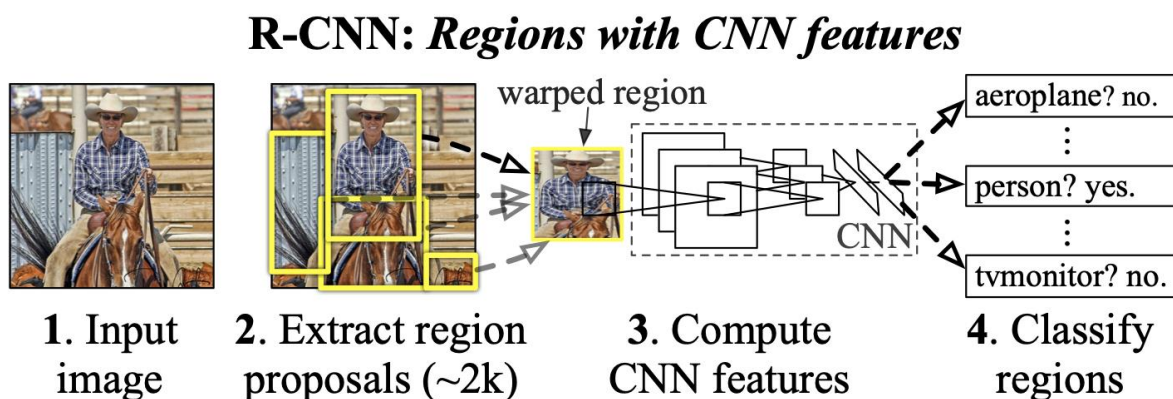


Рисунок 3. Схема работы модели R-CNN. Источник [21]

Работа модели делится на три основных этапа:

(1) Выделение методом selective search [14] 2000 регионов разных размеров и форм, которые позже обрезаются или растягиваются/сжимаются в изображения размером 227x227. Фиксированный размер изображения нужен, потому что после сверточных слоев в архитектуре стоят полносвязные, а ожидают они вектора фиксированного размера.

(2) Далее эти изображения передаются в AlexNet [15] - подобную архитектуру, которая на выходе дает 4096-мерный вектор с выученными признаками изображений.

(3) Наконец, эти векторы проходят через N разных SVM [16] (где N – количество классов и каждая модель обучена на классификацию одного класса). Для предсказания рамок области детекции обучали 4 линейные регрессии (по одной для каждого целевого параметра: две координаты центра рамки и два числа, отвечающие за ширину и высоту). У каждой такой модели входные данные - векторы признаков изображения полученные после CNN и параметры сгенерированной рамки, а целевые значения – это то, насколько нужно изменить эти параметры (насколько сместить центр, и насколько растянуть рамки по каждой оси). В итоге имеем 5 значений: предсказанный класс и 4 значения того, насколько нужно изменить предложенный бокс.

После всех этих операций получается $K * N$ боксов (где K – количество изначально предложенных регионов методом selective search, N – количество классов). Очевидно, что такое количество предсказаний избыточно, поэтому

получившиеся рамки отфильтровывают методом NMS (Non Maximum Suppression).

CNN предобучали на ILSVRC2012 [18]. Так как изначально модель нужно было научить решать задачу детекции по вырезанным частям изображения, то ее дообучали на работу с такими входными данными, сменив полносвязные слои на аналогичные, но с выходами на $20 + 1$ (в случае соревнования PASCAL VOC [17]) и $200 + 1$ (в случае соревнования ILSVRC2013) классов. Единица прибавляется, в силу появления нового класса "задний фон", который является правильным ответом тогда, когда ни один из существующих классов не подошел.

Во время дообучения все регионы, которые имели IoU с истинным боксом больший 0.5 помечались положительным лейблом этого класса, остальные – отрицательным.

В экспериментах, описанных в данной работе, отмечается, что если не дообучать модель на регионах из оригинального изображения, а просто использовать предобученную CNN, то полносвязные слои оказываются практически бесполезны. Отказавшись от всех полносвязных слоев, модель не сильно теряет в метрике. Однако, если дообучить модель, как это делалось изначально, то полносвязные слои начинают играть ощутимо большую роль в финальном решении модели.

3.2.SPP-NET (Spatial Pooling Pyramid Network)

В данной работе авторы выдвинули [7] несколько интересных идей по улучшению R-CNN модели.

Во-первых, в R-CNN каждый из 2000 предложенных регионов изображения проходит через глубокую нейронную сеть. Этот процесс, становится причиной существенных временных затрат. В новом подходе сначала генерируются предложенные боксы, запоминаются их локации, далее все изображение целиком пропускается через CNN и из полученных feature map уже извлекаются определенные значения, соответствующие исходно-предложенным боксам. Результат один и тот же – имеем C (где C – количество каналов в последнем сверточном слое) feature map, которые описывают предложенный регион в признаковом пространстве.

Во-вторых, авторы статьи утверждают, что обрезание предложенных регионов и их растягивание для того, чтобы входные данные у R-CNN имели фиксированный размер, мешает точной детекции. Подобные изменения могут не помешать классификации, ведь относительное расположение деталей изображения не меняется, но масштаб изображения изменится. При

растягивании/сжатии части объекта, сам объект может показаться шире/уже, и это скажется на точности рамок. Однако авторы заметили, что требование фиксированного размера изображения исходит не от сверточных слоев, а от полносвязных. У сверточных слоев вход может быть любой, ведь условно сеть скользящим окном проходит по всему изображению фиксированными шагами, однако из-за этого выход после слоя будет переменным, а полносвязные слои ожидают фиксированное количество значений.

Чтобы решить эту проблему, был предложен новый Spatial Pooling слой, который и дал название всей архитектуре.

Работа слоя заключается в следующем: какого бы размера не получилась конечная feature map, пройти по нему макс-пулингами с разного (но фиксированного) размера фильтрами, а полученные значения по очереди складывать в одномерный вектор. Размеры окон SP слоя рассчитываются таким образом чтобы в конечном векторе всегда было одинаковое количество значений. И тогда входными изображениями для CNN могут быть изображения любого размера (для удобства все еще предполагается равенство высоты и ширины, но этот слой можно обобщить на изображения с произвольным соотношением сторон). Схема работы SPP слоя изображена на рисунке 4.

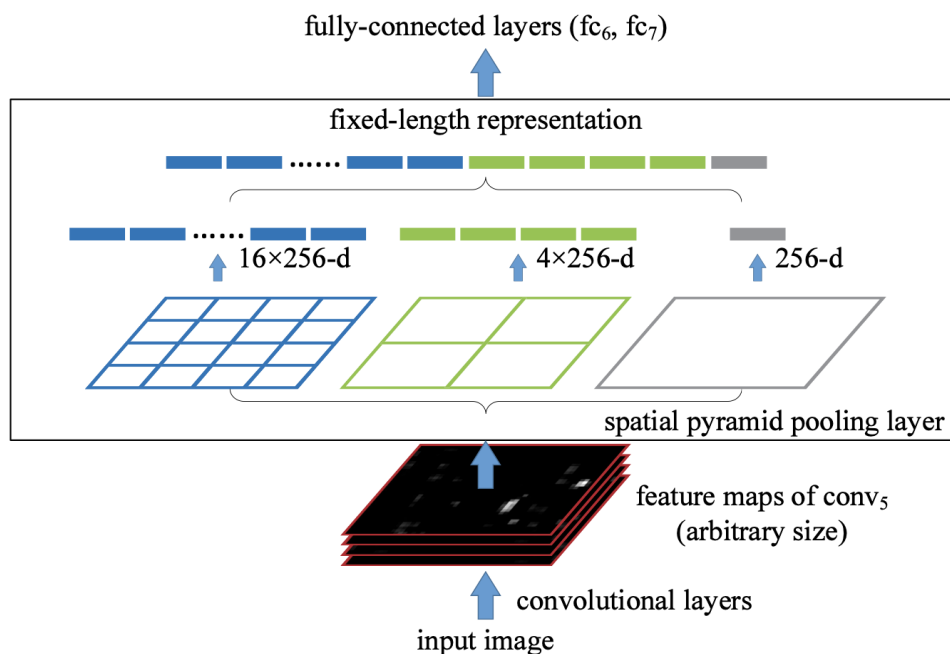


Рисунок 4. Схема работы SPP слоя. Источник [24]

В реальности, обучить модель на изображениях любого размера не удалось в силу ограниченности эффективных реализаций нейросетей на GPU (таких как *cuda-convnet* и *Caffe*). Поэтому авторы обучали модель на двух размерах изображений 180×180 и 224×224 , поочередно меняя размер входного изображения для каждой эпохи. По факту это были две разные нейросети, но которые каждую эпоху передавали друг другу веса и обучались по очереди.

Стоит отметить, что обучение происходило так же, как в R-CNN за исключением, того, что все сверточные слои до SP-слоя не обновляют веса, так как дифференцирование функции Spatial Pooling не представляется возможным.

3.3. Fast R-CNN

Авторы данной статьи своей новой архитектурой [9] решают проблему того, что как в SPP-Net, так и в R-CNN, обучение модели – все еще не однородный процесс. Другими словами, у нас отдельно тренируются N (по количеству классов) разных SVM, отдельно тренируются 4 регрессора для параметров рамок, отдельно дообучается CNN, везде разные функционалы ошибок, а в некоторых свертках SPP-NET слоя вообще веса не обновляются при обратном распространении ошибки.

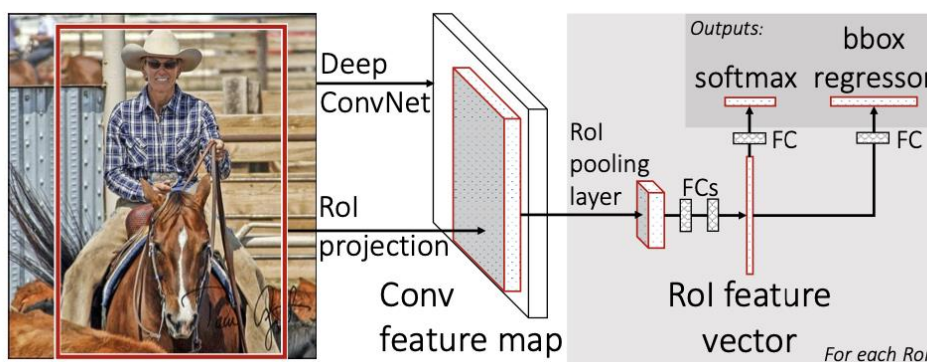


Рисунок 5. Архитектура Fast R-CNN. Источник [25]

Была предложена архитектура, изображенная на рисунке 5. Сначала методом selective search предлагаются 2000 потенциально-интересных регионов, их местоположения запоминаются. Далее, изображение целиком пропускается через CNN (сохраняя при этом исходный размер изображения с помощью нужных отступов при свертках), и поканально выделяются определенные части feature map изображения (части совпадают с местоположениями, полученными при выделении регионов при selective search). До этого момента – все как в SPP-Net. Со следующего шага начинаются изменения.

Теперь вместо SP слоя – новый RoI Pooling слой (Regions of Interest). Каждый предложенный регион изображения (region of interest) на финальной feature map делится на K (K – гиперпараметр) равных квадратных (настолько, насколько это возможно, лишние части отбрасываются) частей и каждая из этих частей проходит через max pooling, после чего мы имеем ровно K чисел, вне зависимости от того, какого размера был входной регион. Таким образом мы сохраняем возможность, использовать regions of interest любого размера,

но при этом не теряем возможность обратного распространения ошибки, как это случилось в SPP-NET.

Далее, полученные вектора проходят через пару полносвязных слоев и получившийся после них вектор идет на вход как софтмаксу для классификации, так и в регрессор изменений параметров предложенного бокса. Однако, ключевая разница заключается в том, что при такой архитектуре оптимизируется функционал общей ошибки, являющейся суммой ошибки классификации и регрессии с каким-то коэффициентами. В архитектурах, указанных ранее, каждая часть модели тренировалась абсолютно обособленно.

4. Подходы к решению поставленной задачи

Задачу распознавания символов на табличках можно сформулировать несколькими способами. Основной переменной, не позволяющей получить однозначную постановку задачи, является тип входных данных.

С данными о клинописных табличках можно работать в нескольких видах. Таблица может быть представлена в виде полноценного изображения (Рисунок 1(а)), или как 3D карту, однако второй способ в данной работе рассматриваться не будет, потому что не было найдено выборок с подобным видом данных.

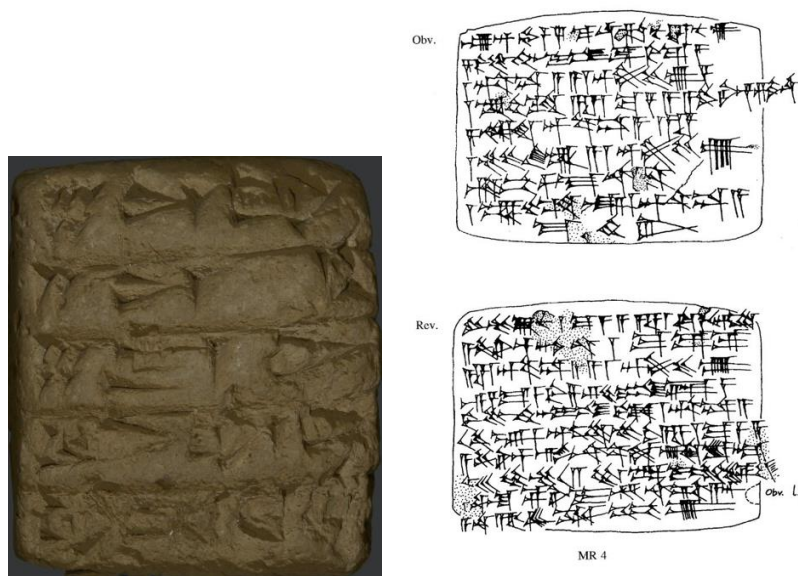


Рисунок 6(а, б). Клинописная таблица и пример транслитерации

Задачу распознавания символов на клинописных таблицах предлагается решать одним из двух способов:

Первый способ заключается в сведении текущей задачи к модифицированной задаче детекции объектов на изображении.

Второй способ заключается в решении задачи на основе графов.

4.1. Задача распознавания символов как задача детекции

Если выбирать в качестве инструмента решения данной проблемы – глубокие нейронные сети, то самым очевидным решением является свести задачу к детекции объектов на изображении, где объектами являются символы.

Исходные изображения все равно не получится обучить модель по нескольким причинам.

Во-первых, мы не получаем преимущество от того, что показываем модели всю табличку целиком. При детекции обычных объектов в их естественной среде существования – это необходимость, потому что задний фон может дать много полезной информации, в случае табличек же - фон нас не интересует, поэтому прежде, чем передавать изображение в модель нужно избавиться от фона.

Во-вторых, на самой табличке много лишней информации. Например, нас не интересует цвет изображения, тени, шероховатости поверхности, которые явно не были нанесены клином. Все это отрицательно скажется на скорости сходимости функционала ошибки. Поэтому, еще одним шагом по предобработке изображения будет некое подобие процессу транслитерации. Результат работы после процесса транслитерации показан на рисунке 7.

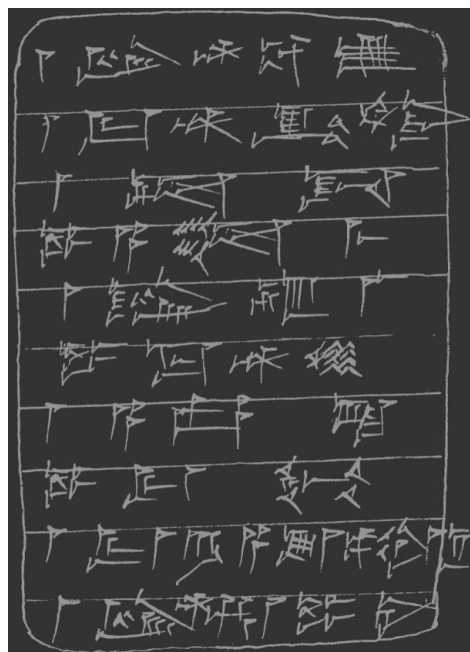
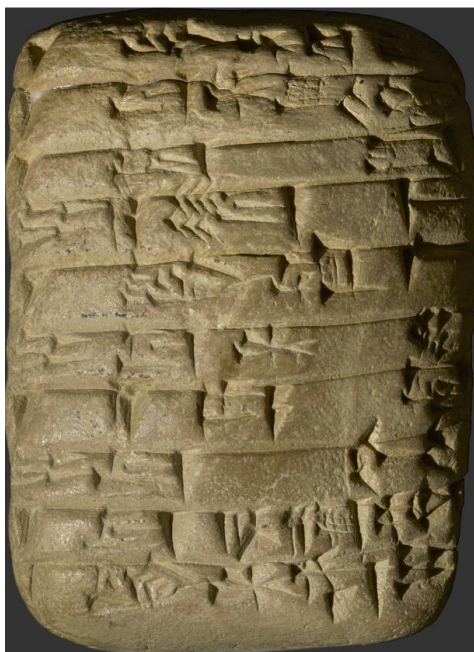


Рисунок 7. Пример транслитерации

Для такой предобработки можно использовать подход Holistically-Nested Edge Detection [1], построенный на полностью сверточной нейронной сети, или Canny edge detection [2]. Результат работы первого алгоритма показан на рисунке 3.

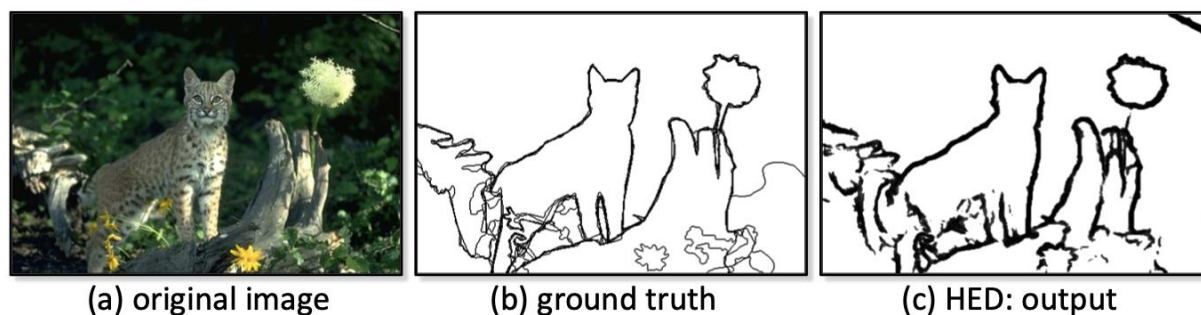


Рисунок 8. Пример работы алгоритма Holistically-Nested Edge Detection

Хочется отметить, что уже на данном этапе работы предложенной модели, значительно упрощается работа людей, занимающихся интерпретацией клинописных табличек. В настоящее время транслитерацией занимаюсь специалисты и каждый подобный перевод является ручным трудом. Предложенный алгоритм позволит автоматизировать этот процесс.

После того, как транслитерации табличек получены, можно приступить к задаче детекции. Предлагается использовать одну из моделей DetectoRS [19], как новую модель (статья 3 июня 2020 года), показывающую наилучшие результаты в области детекции, или Mask R-CNN [18]. В целом – выбор оптимальной архитектуры можно считать отдельным исследованием, и этим можно заниматься уже после того, как будет рабочая модель на основе любой архитектуры.

Модель получает на вход транслитерацию, а на выходе предоставляет то координаты рамок детекции и определяет класс объекта в этих рамках (в нашем случае классом будет являться буква/слово).

4.2. Задача распознавания символов на основе сравнения графов

В направлении данного подхода решения задачи уже имеется несколько исследований [4] [5] [6]. Все эти исследования очень похожи между собой и имеют общую основную идею. Каждый символ представляется не как 2D изображение, а как граф.

При построении сторон и вершин графа будет учитываться расстояние между символами и глубина выреза. Классификация символов будет сведена к классификации графов.

Основную долю описанных статей составляют эвристические подходы к формированию весов на графах, то есть где-то учитывается расстояние до ближайшего, где-то количество смежных сторон, где-то уникальность графа (то есть насколько вообще вероятно встретить такой граф среди множества построенных) и так далее.

Пример построения такого графа по изображению показан на рисунке 4.

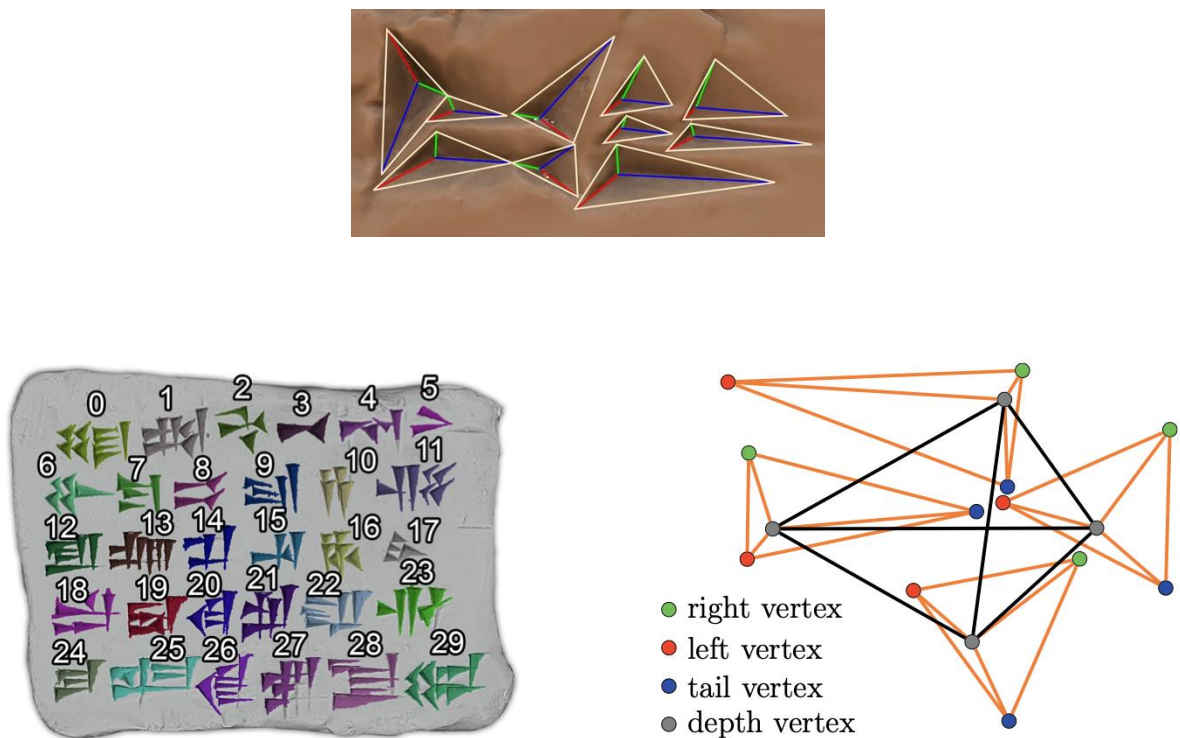


Рисунок 9. Построение части графа по изображению таблицы. На правом нижнем изображении показан граф, формирующийся как в статье [1] для символа под номером 2

4.3. Задача распознавания символов как задача обработки естественного языка

Данный подход является наиболее сложным и требует самых больших вычислительных мощностей, однако именно от него ожидаются наилучшие результаты.

Основой для этого решения являются алгоритмы перевода текста с изображения в обычный текст. Подобные алгоритмы используются в извлечении текстов со сканов, переводе текста с фотографии и частично в распознавании текстов с фотографий вывесок [10] [11] [12].

Архитектура сети при таком подходе делится на две части:

Первая часть состоит из CNN, которая переведет исходное изображение в векторное пространство признаков.

Вторая часть состоит из рекуррентной, которая получит вектор и сгенерирует по нему текст.

Стоит коротко обозначить, как обучаются подобные сети. В определенную архитектуру загружается слово, и в качестве целевой переменной подается все оставшееся предложение, за исключением указанного слова. Существует и обратный подход, при котором в модель загружается предложение без одного слова, и от модели ожидается предсказание того, какое слово было пропущено.

Под капотом, алгоритм считает условные вероятности нахождения каждого слова рядом с другим, и таким образом примерно понимает какую структуру имеют предложения на выбранном языке. Это позволяет не задавать правила языка явными правилами.

Именно поэтому от этого алгоритма ожидаются наилучшие результаты, потому что, как было выяснено во время исследования, один и тот же символ при разных контекстах может иметь разное значение. Простому человеку эта разница не видна, однако считается, что алгоритм может восстановить любую зависимость, если предоставить ему достаточное количество данных.

Этот же принцип решает проблему того, что в клинописных табличках отсутствует четкая структура, нет четкого обозначения строчек, столбцов, абзацев, что сильно усложнило бы жизнь человеку, однако станет лишь незначительной сложностью для алгоритма.

5. Направление будущих исследований

Собрать, очистить и разметить данные

Обучить каждый из предложенных вариантов, выбрать оптимальный

Экспериментировать с подбором гиперпараметров (методов расчета весов на графах, архитектур CNN и архитектур NLP)

6. Заключение

В данной работе были разобраны классические алгоритмы детекции, рассмотрены существующие подходы и предложены новые алгоритмы для решения задачи перевода клинописных табличек.

А также предложены варианты продолжения данного исследования

7. Список литературы

1. Xie, Saining, and Zhuowen Tu. "Holistically-nested edge detection." *Proceedings of the IEEE international conference on computer vision*. 2015.
2. Canny, John. "A computational approach to edge detection." *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986): 679-698.
3. D. Fisseler, F. Weichert, M. Cammarosano, and G. Müller. Towards an interactive and automated script feature analysis of 3D scanned cuneiform tablets. In *Scientific Computing and Cultural Heritage*, 2013.

4. B. Bogacz, M. Gertz, and H. Mara. Character retrieval of vectorized cuneiform script. In International Conference on Document Analysis and Recognition, 2015b.
5. Recognizing Cuneiform Signs Using Graph Based Methods
6. He, Kaiming, et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition." *IEEE transactions on pattern analysis and machine intelligence* 37.9 (2015): 1904-1916.
7. Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.
8. Girshick, Ross. "Fast r-cnn." *Proceedings of the IEEE international conference on computer vision*. 2015.
9. Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems*. 2015.
10. Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems*. 2017.
11. Shi, Baoguang, Xiang Bai, and Cong Yao. "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition." *IEEE transactions on pattern analysis and machine intelligence* 39.11 (2016): 2298-2304.
12. Zacharias, Ebin, Martin Teuchler, and Bénédicte Bernier. "Image Processing Based Scene-Text Detection and Recognition with Tesseract." *arXiv preprint arXiv:2004.08079*(2020).
13. Uijlings, Jasper RR, et al. "Selective search for object recognition." *International journal of computer vision* 104.2 (2013): 154-171
14. A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.
15. Suykens, Johan AK, and Joos Vandewalle. "Least squares support vector machine classifiers." *Neural processing letters* 9.3 (1999): 293-300.
16. Everingham, Mark, et al. "The pascal visual object classes (voc) challenge." *International journal of computer vision* 88.2 (2010): 303-338.
17. Deng, J. et al., 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. pp. 248–255.
18. He, Kaiming, et al. "Mask r-cnn." *Proceedings of the IEEE international conference on computer vision*. 2017.
19. Qiao, Siyuan, Liang-Chieh Chen, and Alan Yuille. "DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution." *arXiv preprint arXiv:2006.02334* (2020).