# АЛГОРИТМ АВТОМАТИЧЕСКОГО ОПИСАНИЯ ИЗОБРАЖЕНИЙ

*Астрашаб В.В.*

*Белорусский государственный университет информатики и радиоэлектроники*
*г. Минск, Республика Беларусь*

*Калугина М.А. – канд. физ.-мат. наук, доцент*

В докладе рассмотрен алгоритм автоматического описания изображений, работающий на основе нейронных сетей. В частности, приведено общее описание принципов его реализации, проанализированы полученные результаты и перспективы его применения.

Автоматическая генерация описаний для изображений является важной задачей, успешное решение которой позволит сэкономить человеческие ресурсы. Правилом хорошего тона является сопровождать все изображения, размещённые на веб-сайтах, текстовым описанием. Это делается по нескольким причинам. Во-первых, это упрощает пользование интернетом и доступ к информации людям с проблемами со зрением, так как текстовая информация может быть озвучена голосовым помощником, а изображения зачастую содержат важную для понимания сути написанного информацию. Во-вторых, этот текст может быть отображён вместо изображения если возникли проблемы с его загрузкой. На данный момент большая часть таких описаний пишется вручную или вовсе игнорируется необходимость их написания. Но на некоторых сайтах, таких как новостные ресурсы, интернет-энциклопедии и другие подобные сайты, изображений содержится так много, что на их описание тратится много времени авторов статей. Кроме того, необходимость описывать изображения возникает в учебниках, технической литературе, научных статьях. Алгоритмы автоматической генерации описаний помогли бы сэкономить время авторам. Даже если качество описаний не будет таким высоким, как у человека, перепроверка и корректирование автоматически сгенерированного описания займут меньше времени, чем составление собственного с нуля.

В данной работе рассмотрен один из современных алгоритмов генерации описаний к изображениям, разработанный в 2016 году исследователями университетов Монреаля и Торонто [1]. Данный алгоритм основан на использовании технологии больших данных. Работу по обработке изображения и перевода его содержания в текст совершают несколько нейронных сетей, предварительно обученных для этого на наборе данных, содержащим множество примеров изображений и описаний к ним. Плюсом этого подхода является то, что программисту не нужно знать особенностей синтаксиса и грамматики языка, уметь выделять из изображения необходимые признаки вручную и сопоставлять их со словами. Все эти задачи нейросетевая модель учится совершать самостоятельно с помощью алгоритмов оптимизации. Задача, выполнение которой остаётся за программистом, – это проектирование архитектуры модели, её реализация на одном из языков программирования, а также сбор и подготовка необходимого набора данных. Последний пункт долгое время был наиболее трудным в осуществлении, так как сложные модели для качественного обучения требуют огромного набора данных, а также ресурсов для их хранения и обработки. В последнее время разработки в сфере больших данных сделали эту задачу осуществимой, поэтому алгоритмы такого типа завоёвывают популярность в сравнении с менее требовательными к данным, но и менее точными и более сложными в разработке классическими алгоритмами.

Реализованная нами модель состоит из трёх нейронных сетей: кодирующей нейронной сети, декодирующей нейронной сети, а также нейронной сети с вниманием.

Задача кодирующей сети состоит в переводе изображения в сжатый вектор признаков, по которым в дальнейшем будет осуществляться составление описания. В качестве кодирующей сети можно брать любую архитектуру для работы с изображением, например, наиболее популярные свёрточные нейронные сети. Для данной архитектуры была выбрана сеть ResNet-101 [2], состоящая из 101 свёрточного слоя. Эта сеть уже была настроена классифицировать изображения, поэтому умеет выделять некоторые признаки, но ее можно дополнительно обучить решать конкретную задачу. Такой подход позволяет сети обучаться быстрее, так как не приходится делать это с нуля.

Задача декодирующей сети состоит соответственно в генерации текста по выделенным признакам. Для этого используется рекуррентная нейронная сеть с LSTM-ячейками памяти [3]. Получая вектор признаков, она генерирует по нему слово и некоторый вектор скрытого состояния. Этот вектор она передаёт в точно такую же ячейку. Процедура повторяется до тех пор, пока сетью не будет сгенерирован символ конца предложения. Также для генерации каждого слова декодирующая сеть принимает на вход результат работы сети с вниманием.

Сеть с вниманием работает совместно с декодирующей сетью. Она получает на вход предыдущее сгенерированное слово, а также признаки, выделенные кодирующей сетью. Её задача заключается в том, чтобы выделить наиболее важные из признаков для генерации следующего слоя. Это позволяет модели «обращать внимание» на различные признаки попеременно, наподобие того, как человек рассматривает изображение, последовательно замечая различные объекты на нём. Данная сеть была предложена исследователем Монреальского университета, выпускником факультета прикладной математики и информатики БГУ Дмитрием Богдановым [4]. Изначально эта архитектура использовалась для задач машинного перевода, но впоследствии нашла своё применение и в других задачах.

Все нейронные сети из приведённой архитектуры затем обучаются совместно. Обучение проводилось на наборе данных MSCOCO '14, содержащим более 200 тысяч размеченных изображений и находящимся в открытом доступе для использования в исследовательских целях. Каждое изображение из данного набора данных имеет своё описание на английском языке. Каждое изображение было сжато нами до 256x256 пикселей. Общий вес набора данных составляет около 19 Гб.

Обучение проходит на тренировочном наборе данных, а проверка регулярно осуществляется на наборе данных для валидации. Это позволяет отследить эффект переобучения, когда модель слишком сильно приспосабливается к тренировочным данным и теряет способность обобщения на новые данные. Это можно заметить с помощью валидации, если во время обучения качество модели на тренировочных данных растёт, а на валидационных – убывает. Для предотвращения проблемы переобучения используется регуляризация. В данной модели используется один из способов регуляризации – накладывание ограничения на величину параметров модели.

Для обучения используются алгоритмы оптимизации, а оптимизируется функция потерь, в качестве которой используется перекрёстная энтропия.

Для реализации данной модели мы использовали язык программирования Python 3.6 с библиотекой для глубокого обучения TensorFlow и множеством вспомогательных библиотек. Выбор данного языка и библиотек обусловлен наличием в них наиболее широкого набора инструментов для разработки моделей машинного обучения.

Пример результатов работы алгоритма приведён на рисунке 1. Над изображением содержится описание, сгенерированное с помощью алгоритма на английском языке.

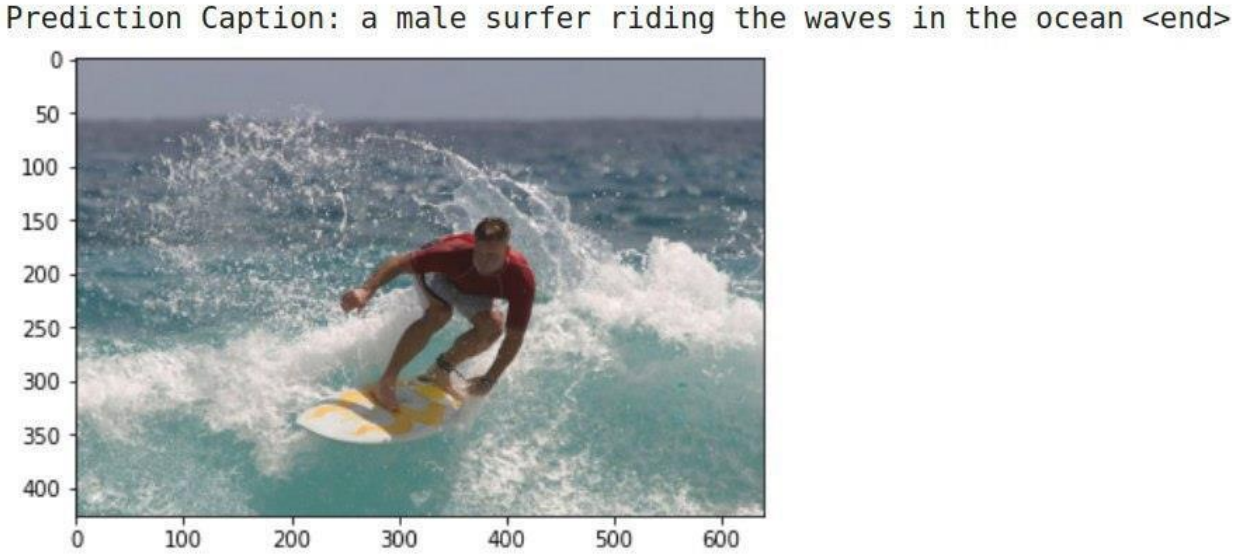


Рисунок 1 – пример работы модели на изображении

Таким образом, в результате проведённой работы, был изучен и реализован алгоритм автоматической генерации описания к изображениям. При наличии соответствующего набора данных сфера применения данного алгоритма может быть расширена. В частности, ту же самую архитектуру можно использовать для автоматического составления описаний на любых языках, а не только на английском. Набор данных может быть заменён на более подходящий к специфичной области применения для улучшения работы алгоритма на изображениях в данной области. Если такой набор данных достаточно велик и существенно отличается от предложенного в работе, можно провести обучение модели с нуля, иначе можно использовать уже обученную модель и дообучить её для более конкретной задачи. Работает данный алгоритм с большой скоростью (менее секунды на одно изображение), что позволяет быстро генерировать описание для множества изображений.

***Список использованных источников:***

1. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention / K. Xu [et al.] // arXiv:1502.03044v3 [cs.LG], 2016.
2. Deep Residual Learning for Image Recognition / K. He [et al.] // arXiv:1512.03385v1 [cs.CV], 2015.
3. Long Short-term Memory / S. Hochreiter, J. Schmidhuber // Neural Computation 9(8):1735-80, 1997
4. Neural Machine Translation by Jointly Learning to Align and Translate / D. Bahdanau [et al.] // arXiv:1409.0473v7 [cs.CL], 2016