

**РАСПОЗНАВАНИЕ ЯПОНСКИХ ИЕРОГЛИФОВ
С ПОМОЩЬЮ МЕТОДОВ МАШИННОГО
И ГЛУБОКОГО ОБУЧЕНИЯ**

Аннотация. В статье рассмотрено использование методов синтезирования обучающей выборки, а также методов машинного и глубокого обучения для распознавания японских иероглифов.

Ключевые слова: японские иероглифы, машинное обучение, глубокое обучение, нейронные сети.

Введение

В данной работе рассмотрена разработка приложения для распознавания основных японских иероглифов.

В настоящее время становятся популярными поездки за границу, причем нередко туристы не знают язык страны, в которую они отправляются. С современными возможностями использования онлайн-переводчиков это не является проблемой, однако поиск в переводчике каждого незнакомого слова может отнять много времени, особенно при незнании алфавита. Несложно представить, сколько красочных вывесок будет окружать туриста. Каждая из них может содержать полезную информацию, понимание которой поможет найти гостиницу, продуктовый магазин или аптеку. Таким образом, туристу было бы удобнее ориентироваться, если бы он имел возможность быстро перевести надпись на очередной вывеске.

Возникает необходимость перевести вывеску без использования интернета, а также без необходимости носить с собой словарь. Японские иероглифы уникальны для каждого слова, что создает трудности в их

запоминании. Вот почему было решено реализовать быстрое распознавание иероглифов с помощью методов машинного и глубокого обучения.

Целью данного проекта было проверить, можно ли использовать нейронные сети для распознавания на изображениях основных японских иероглифов.

Обзор

При подготовке к обучению нейронной сети возникает проблема создания обучающей выборки, так как существующие обучающие выборки не содержат японских иероглифов. Набор рукописных изображений был подготовлен с помощью графического редактора Paint из стандартного набора приложений MS Windows.

Для увеличения количества вариантов созданных рукописных изображений были применены обработчики, реализующие размытие, искажение с помощью преобразования перспективы, сдвига, поворота исходного изображения.

Для размытия изображений был использован фильтр Гаусса, использующий нормальное распределение для вычисления преобразования, применяемого к пикселям изображения [1]. В приведенной статье рассмотрена реализация алгоритма размытия по Гауссу.

Искажение изображений может быть реализовано через преобразование перспективы, что позволяет изменить отдельные элементы иероглифа, вытянув их или наоборот сжав. Таким образом, создается имитация разных вариантов написания иероглифа. Был изучен электронный ресурс с обсуждением реализации преобразования перспективы на языке программирования Python с использованием библиотеки Pillow [2].

Для выбора методов реализации нейронной сети для распознавания основных японских иероглифов были изучены статьи с похожей тематикой.

В руководстве, предложенном TensorFlow [3], для написания нейронной сети использованы язык программирования Python 3, библиотеки TensorFlow, Keras, набор данных Fashion MNIST. На данном ресурсе

детально разобран принцип составления модели для распознавания одежды на изображениях. Для обучения нейронной сети используется 60.000 изображений, которые она будет распределять по десяти классам, а также тестовый набор из 10.000 изображений, на которых будет оцениваться точность обученной модели. Также на данном сайте рассмотрено, в каком формате данные должны передаваться в нейронную сеть, что тоже необходимо для решения поставленной задачи. В руководстве рассмотрено создание нейронной сети из трех слоев, на выходе обученная модель выдает номер класса, к которому по ее предсказанию относится изображение. Данная задача относится к распознаванию объектов на изображениях, что является схожей задачей с распознаванием иероглифов.

Основной сложностью распознавания иероглифов является то, что для обучения нейронной сети необходимо самостоятельно собрать набор данных с изображениями и метками, соответственно, важно понимать, как эти данные будут загружаться в нейронную сеть и как их для этого подготовить.

В другом руководстве по TensorFlow, предоставленном Google Developers Codelabs [4] также рассмотрено создание нейронной сети для распознавания объектов на изображениях, в данном примере – распознавания цифр. Для данной задачи были выбраны язык программирования Python 3, библиотека TensorFlow и набор данных MNIST, содержащий 70.000 изображений, как и в предыдущем примере, разделенных на набор для обучения и тестовый набор. Данный ресурс предоставляет подробное описание работы с библиотекой TensorFlow на наглядном примере, что позволяет по шагам разобрать, как строится нейронная сеть, как происходит взаимодействие между ее слоями, что влияет на точность распознавания.

В настоящей работе использованы следующие подходы: язык программирования Python 3, библиотеки Keras (написание нейронной сети), Pillow (работа с изображениями, преобразование цветовых значений пикселей в числовые значения), NumPy. Для обучения нейронной сети была подготовлена обучающая выборка из 20.550 изображений, которые должны

быть распределены по 10 классам (иероглифы-цифры). На вход в первый слой нейросети подается массив изображений, преобразованных в черно-белый формат. Цветовые значения пикселей представлены числами от 0 до 1. В качестве меток используются одномерные массивы, заполненные нулями и одной единицей, индекс которой соответствует классу, к которому принадлежит изображение.

Материалы и методы решения

Сформирована обучающая выборка. Были взяты изображения с рукописными иероглифами и преобразованы к формату 32*32 пикселей. Для имитации возможных искажений на реальных фотографиях с иероглифами применен обработчик преобразования перспективы, результаты работы которого представлены на рисунке 1.



Рис. 1. Преобразование перспективы.

Угловым точкам изображения ставятся в соответствие новые точки, две соседние из которых остаются без изменений, а две другие сдвигаются на четверть расстояния между ними. Пиксели между четырьмя угловыми преобразуются в соответствии с указанным правилом.

Принцип преобразования перспективы был так же использован для написания обработчиков, реализующих поворот изображения и сдвиг. Поворот – наклон изображения под определенным углом. Для формирования

обучающей выборки было решено использовать поворот, не более чем на 10° . Для совершения такого преобразования исходным точкам изображения ставятся в соответствие новые точки, смещенные на некоторое расстояние относительно исходных координат так, чтобы расстояния между каждыми двумя соседними точками были равны. Результаты преобразований приведены на рисунке 2.



Рис. 2. Поворот изображения.

Сдвиг изображения – две соседние угловые точки сдвигаются на одно и то же число пикселей в одну сторону, а две другие точки – на это же число пикселей в противоположную сторону. Пример такого преобразования представлен на рисунке 3.



Рис. 3. Сдвиг изображения.

Для имитации размытых фотографий используется фильтр Гаусса, преобразующий цветовые значения пикселей в соответствии с цветовыми значениями соседних к ним пикселей, из-за чего образуется размытие. Пример представлен на рисунке 4.



Рис. 4. Фильтр Гаусса.

При загрузке изображений в нейронную сеть происходит преобразование цветовых значений пикселей к диапазону от 0 до 1.

Реализована нейронная сеть с двумя полносвязными слоями и двумя сверточными слоями с использованием языка программирования Python и библиотеки Keras, для оценки точности работы нейронной сети применена метрика Ассигасу. В качестве функции активации выбрана функция Relu, которая наиболее часто используется при распознавании объектов на изображениях.

$$y = \max(0, x)$$

Для загрузки обучающей и тестовой выборки в нейронную сеть использован ImageDataGenerator из библиотеки Keras, что позволяет автоматически преобразовать поступающие изображения к нужному формату, когда цветовое значение каждого пикселя находится в промежутке от 0 до 1 и исходное изображение преобразовано в матрицу.

Необходимость использовать ImageDataGenerator возникает из-за того, что легко допустить ошибку при преобразовании цветовых значений пикселей в числа от 0 до 1, что потом оказывает влияние на обучение нейронной сети. При распознавании 10 иероглифов-цифр (рис. 5) точность классификации обученной нейросети составляла только 10%, а после использования ImageDataGenerator выросла до 90%.

1	2	3	4	5	6	7	8	9	10
一	二	三	四	五	六	七	八	九	十

Рис. 5. Иероглифы-цифры

Также на точность распознавания нейронной сети влияет качество обучающей выборки. При сильном искажении изображений точность распознавания обученной модели снижается.

Результаты

Точность распознавания обученной нейросетью тестовой выборки из 500 изображений на каждый из 10 иероглифов-цифр составила 100%.

Несмотря на то, что была получена высокая точность распознавания, этот результат ухудшается при работе с большим количеством классов, чем использованные 10 цифр. Следует также отметить, что иероглифы-цифры состоят из простого набора элементов и сильно отличаются друг от друга, однако в японском языке много иероглифов с более сложной структурой, а также многие иероглифы имеют общие элементы, что усложняет их распознавание [5].

Заключение

В дальнейшем для классификации иероглифов по большому числу классов можно рассмотреть выделение похожих элементов у иероглифов и кластеризацию по этим признакам.

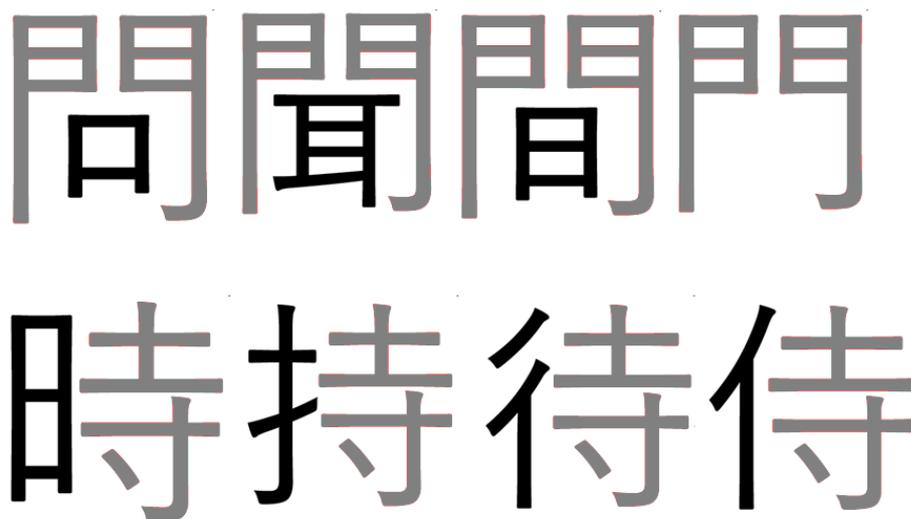


Рис. 6. Примеры иероглифов с одинаковыми элементами.

На рисунке 6 представлены примеры похожих иероглифов, содержащих одинаковые элементы. При классификации иероглифов по большому числу классов можно добиться более высокой точности, если обучить модель выделять признаки, характерные для групп иероглифов, а потом проводить классификацию внутри этих групп.

В перспективе обученную нейронную сеть можно применить для разработки приложения-переводчика для быстрого распознавания японских иероглифов по изображениям.

СПИСОК ЛИТЕРАТУРЫ

1. Программная реализация размытия по Гауссу [Электронный ресурс. <http://ftp.altlinux.org/pub/people/at/gblur.pdf>] (дата обращения: 05.05.2019).
2. Преобразование перспективы в PIL [Электронный ресурс. <http://qaru.site/questions/187252/how-does-perspective-transformation-work-in-pil>] (дата обращения: 05.05.2019).
3. Обучи свою первую нейронную сеть: простой классификатор [Электронный ресурс. https://www.tensorflow.org/tutorials/keras/basic_classification?hl=ru] (дата обращения: 05.05.2019).
4. TensorFlow and deep learning, without a PhD [Электронный ресурс. <https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#0>]
5. Список кандзи [Электронный ресурс. <https://akanji.ru/JLPT/N5>] (дата обращения: 05.05.2019).