УДК 528.854 *2.2.8*

ПРОЦЕДУРЫ ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ ДЛЯ ПОИСКА И РАСПОЗНАВАНИЯ ГРУЗОВЫХ КОНТЕЙНЕРОВ

Р.А. Макаров
(научный руководитель: О.Г. Морозов, профессор КНИТУ-КАИ)

Казанский национальный исследовательский технический университет
им. А.Н. Туполева—КАИ
Российская Федерация, 420111, г. Казань, ул. К. Маркса, 31/7

Аннотация. В данной статье описывается обучение нейронной сети Faster Rcnn Resnet 50 для поиска и распознавания грузового контейнера на изображении. Для обучения модели необходимо собрать данные для обучения, а также разметить их соответствующим образом. Обучение нейронной сети производилось на языке программирования Python с использованием библиотеки Tensorflow.

Ключевые слова: изображение, грузовой контейнер, поиск, распознавание, нейронная сеть.

Введение

В последнее время широкое распространение получили смешанные архитектуры нейронных сетей, включающие в себя алгоритм поиска и распознавания объекта на изображении [1]. Использование подобных архитектур нейронных сетей существенно упрощает алгоритм обработки изображений, а также в некоторых случаях позволяет ускорить процесс распознавания объектов на изображении [2].

В данной статье описывается обучение нейронной сети Faster Rcnn Resnet 50 для поиска и распознавания грузового контейнера на изображении. Для обучения модели необходимо собрать данные для обучения, а также разметить их соответствующим образом.

Сбор данных для обучения

Как и любая нейронная сеть, Faster Rcnn Resnet 50 на вход принимает изображение [3]. Необходимость сбора данных обусловлена отсутствием готового набора данных для обучения. Сбор данных можно осуществлять вручную или автоматически, с использованием скрипта для скачивания изображений с ресурсов. При автоматическом сборе изображений - отсутствует контроль качества данных, а также проверка соответствия, что на изображении находится грузовой контейнер, поэтому для решения данной задачи был выбран ручной способ сбора данных. Пример полученных данных для обучения модели представлен на рисунке 1. В результате ручного сбора данных удалось получить набор из 250 изображений, это является недостаточным количеством для обучения модели [4]. Для искусственного расширения набора данных можно применить аугментацию. В данном случае аугментация позволит повысить вариативность и дополнить изображения менее качественными данными для обучения, что в результате повысит точность работы нейронной сети при распознавании объекта.

Рис. 1. Пример собранных данных для обучения модели.

Аугментация и расширение данных для обучения

В связи с недостаточным количеством данных для обучения, было принято решение использовать аугментацию. В качестве наиболее популярных способов аугментации можно выделить следующие:

- отражение по вертикали и горизонтали,
- поворот изображения,
- сдвиг изображения,
- обрезание изображения,
- добавление искусственного шума,
- изменение контрастности и яркости изображения.

Пример использования аугментации представлен на рис. 2. Для этого применялось добавление искусственного шума, изменение яркости и контрастности изображения.

Рис. 2. Пример аугментированного и исходного изображения

Для каждого изображения создавалось 5 новых, с использованием аугментации, это позволило увеличить набор данных с 250 изображений до 1500 изображений.

Разметка изображений для обучения

В связи с тем, что помимо классификации нейронная сеть выполняет функцию по поиску объекта на изображении, разметка обучающего набора данных помимо класса должна включать в себя координаты объекта на изображении [5]. Координаты объекта включают в себя 4 точки на изображении, которые образуют прямоугольник, в котором находится распознаваемый объект. Для разметки изображения удобно использовать утилиту LabelImg. Интерфейс данной утилиты представлен на рис. 3. Он позволяет выделить квадрат с объектом на изображении, а также присвоить один или несколько классов для выделенного объекта.

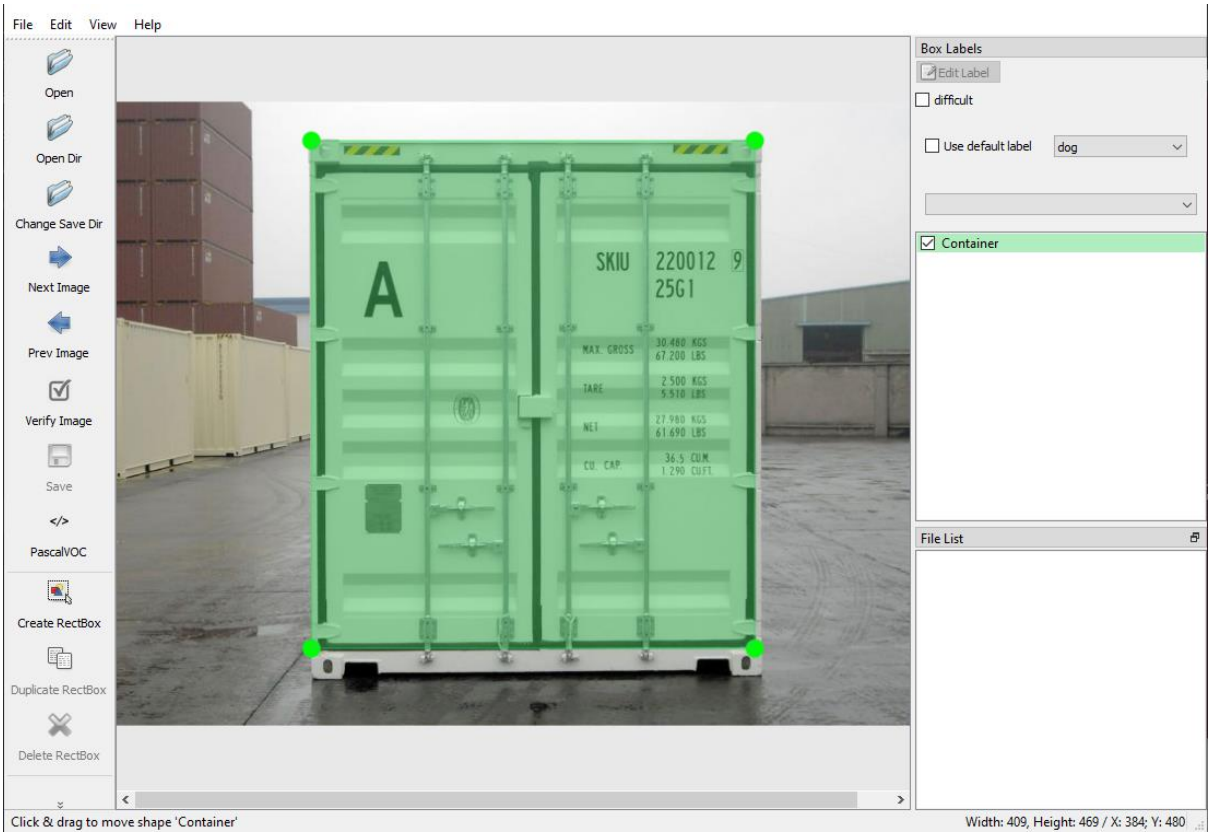

Рис. 3. Интерфейс утилиты LabelImg

В результате для каждого размеченного изображения формируется .xml файл, который содержит следующие данные:

- полный путь до изображения в системе;
- размер изображения в пикселях, а также количество каналов;
- координаты четырех точек выделенного объекта, а также его класс.

Для повышения удобства работы с разметкой, было принято решение преобразовать .xml файлы в единый csv файл.

Помимо непосредственной разметки изображений, набор данных необходимо разделить на тренировочный и тестовый [6]. Отличие данных наборов заключается в следующем: тренировочный набор используется для непосредственного обучения нейронной сети, тестовый - для проверки точности модели между итерациями обучения, а

также для проведения контрольной проверки точности нейронной сети после процесса обучения. В связи с этим исходный набор данных из 1500 изображений был разделен на тренировочный и тестовый в соотношении 85 на 15 соответственно.

Обучение нейронной сети

Обучение нейронной сети производилось на языке программирования Python с использованием библиотеки Tensorflow. Архитектура нейронной сети представлена на рис. 4.

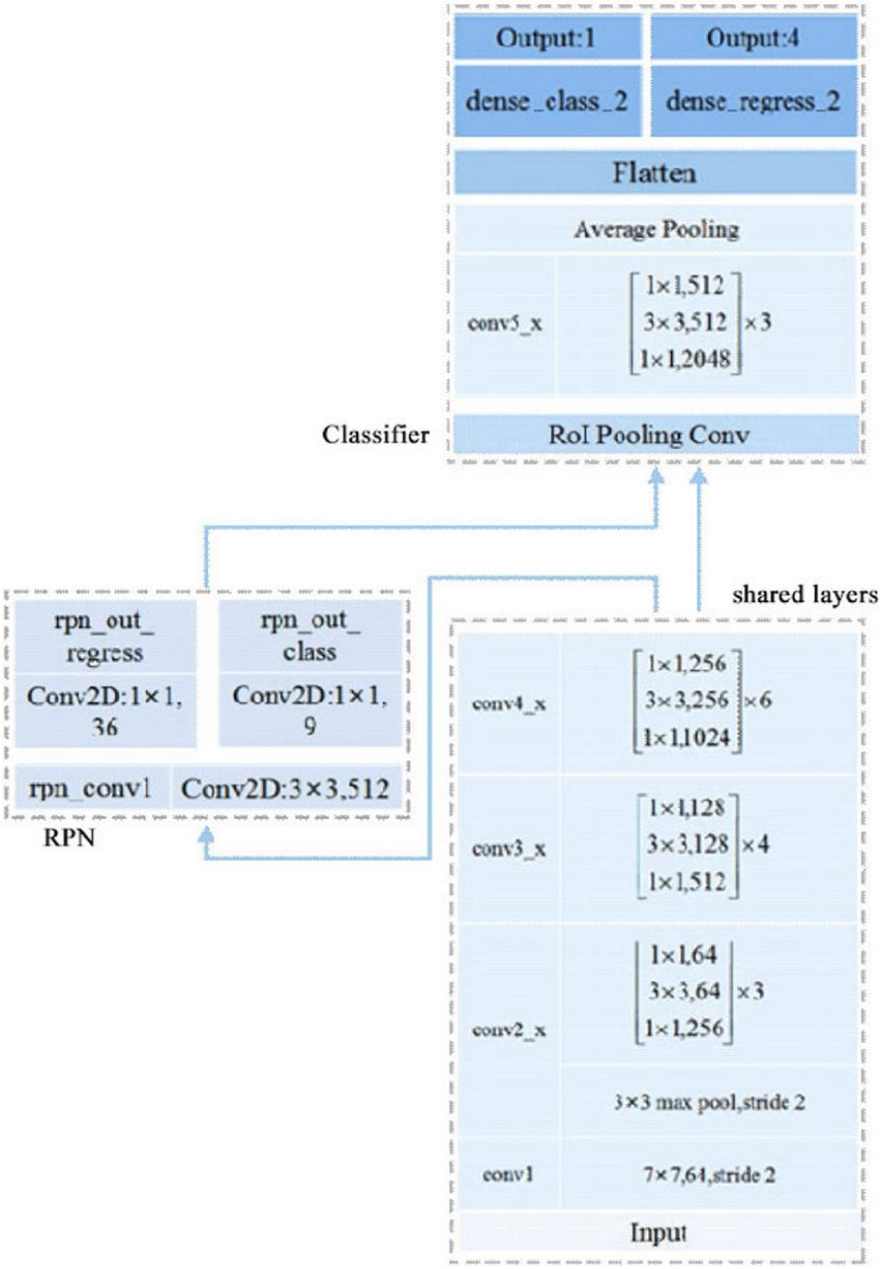

Рис. 4. Архитектура нейронной сети Faster Rcnn Resnet 50

Обучение производилось с использованием GPU(NVidia GTX 1660). Было произведено 16 700 итераций с размером выборки 32, что заняло около 6 часов. График функции потерь тестовой выборки представлен на рис. 5.

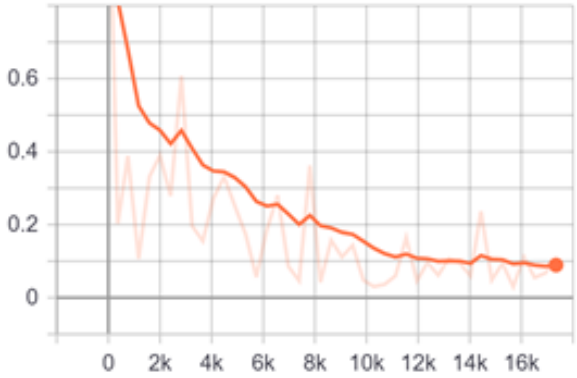

Рис. 5. График функции потерь при обучении модели

Рис. 6. Пример работы модели на тестовых данных

Тестирование модели

Проверка точности модели производилось на размеченных тестовых данных, а именно на 225 изображениях. В качестве показателя точности выделения области с объектом было принято решение использовать коэффициент степени пересечения между двумя ограничивающими рамками. В качестве порога коэффициента пересечения было принято решение взять 0.8, таким образом в случае если коэффициент пересечения меньше 0.8, принимается решения о том, что область с объектом выделена не верно. Коэффициент степени пересечения вычисляется как отношение площади пересечения предсказанной и истинной ограничивающей прямоугольной области к площади объединения предсказанной и истинной ограничивающей прямоугольной области [7]. Схематически данный коэффициент представлен на рис. 7.

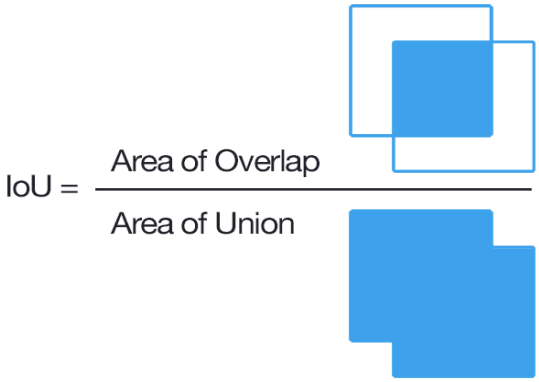

Рис. 7. Схематическое изображение коэффициента степени пересечения.

В результате тестирования, точность классификации объектов составила 94,5%, точность определения области с объектом составила 87,4%.

Заключение

В результате данной работы, была получена обученная нейронная сеть, способную находить и распознавать грузовой контейнер на изображении. В дальнейшем планируется усовершенствовать данную модель с точки зрения повышения стабильности работы нейронной сети, а также расширение выборки тренировочных и тестовых данных.

Список литературы

1. R-CNN, Fast R-CNN, Faster R-CNN, YOLO–Object Detection Algorithms [Электронный ресурс]. – Режим доступа: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e. – (Дата обращения 23.09.2022)
2. Николенко С.И. Глубокое обучение. Погружение в мир нейронных сетей. / С.И. Николенко, А.А. Кадурин, Е.В. Архангельская – Санкт-Петербург: Питер, 2018. – 480 с.
3. Girshick, R. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation / R. Girshick , J. Darrell, T. Malik // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2015. – № 38.
4. Wang, Yuanyuan. Automatic Ship Detection Based on RetinaNet UsingMulti-Resolution / Y. Wang, C. Wang , H. Zhan, G. Yingbo, S. Wei // Remote Sensing. – 2019. – № 11. – С. 531.
5. Ross Girshick. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [Электронный ресурс] – Режим доступа: https://arxiv.org/pdf/1506.01497.pdf (дата обращения 02.10.2022)
6. Rich feature hierarchies for accurate object detection and semantic segmentation. Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik [Электронный ресурс] – Режим доступа: https://arxiv.org/pdf/1311.2524.pdf (дата обращения 19.09.2022)
7. Y. Ren, C. Zhu, and S. Xiao, Small object detection in optical remotesensing images via modified faster R-CNN // Appl. Sci. - 2018. - V.8, N.5. - P. 813.

NEURAL NETWORK TRAINING PROCEDURES FOR SEARCHING AND RECOGNIZING CARGO CONTAINERS

R.A. Makarov
(Supervisor: G.A. Morozov, professor KNRTU-KAI)

Kazan National Research Technical University named after A.N. Tupolev –KAI
10, K. Marx, Kazan, 420111, Russian Federation

Annotation. This article describes training a Faster Rcnn Resnet 50 neural network to search and recognize a cargo container in an image. To train the model, you need to collect training data and label it appropriately. The neural network was trained in the Python programming language using the Tensorflow library.

Keywords: image, cargo container, search, recognition, neural network.

Статья представлена в редакцию 24 сентября 2022г.