МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное

учреждение высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

**Институт компьютерных технологий и информационной безопасности**

**Кафедра Математического обеспечения и применения ЭВМ**

 

**ОТЧЁТ**

по курсовой работе

по курсу «Машинное обучение»

по теме «Семантическая сегментация»

Выполнил:

студент группы КТмо2-3

Шепель И.О.

Проверил:

к. т. н., Гончаров А. В.

Оценка

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2017 г.

Таганрог 2017

1 Введение 3

2 Структура PSPNet 3

3 Архитектура сети 5

4 Экспериментальные результаты 5

Список использованных источников 9

# 1 Введение

Для решения задачи семантической сегментации, т.е. попиксельного выделения объектов на изображениях, используются сверточные сети с различной архитектурой. В частности на наборе данных cityscapes [1], одной из наиболее успешных архитектур является сеть с пирамидальным анализом сцены – PSPNet (англ. – Pyramid Scene Parsing Network)[2].

В отчете рассматривается задача обучения сети с такой архитектурой на наборе данных cityscapes и с использованием предобученной сети ResNet-101 [3].

Набор данных cityscapes, на котором производилось обучение, представляет собой 5000 размеченных попиксельно изображений с 30 различными классами. Все кадры снимались камерой установленной в салоне автомобиля. Поэтому этот набор данных является наиболее удобным для решения задач выделения объектов в тематике беспилотного автотранспорта.

# 2 Структура PSPNet

Полные сверточные сети (англ. – fully convolutional networks) обладают рядом недостатков, которые проявляются на сложных сценах.

Во-первых, при распознавании объекта на сцене не учитывается характер самой сцены. Так, например моторная лодка, может выделиться как машина из-за схожести этих объектов. При этом, если учитывать, что лодка находится на водной поверхности, а машины – на дорожном покрытии, т. е. саму сцену, на которой расположен объект, то таких ошибок классификации можно избежать.

Во-вторых, на наборах данных с большим числом различных классов очень часто некоторые из них описывают схожие объекты. В частности, на наборе данных cityscapes существуют отдельные классы для легковых, грузовых автомобилей и автобусов. При этом полные сверточные сети могут помечать такие объекты частично одним классом, частично другим.

В-третьих, некоторые объекты обладают небольшим размером на изображении, но при этом очень важны для решаемых задач. Например, знаки дорожного движения и светофоры. Сверточная сеть может неправильно классифицировать такой объект, если он находится на фоне более крупного объекта другого класса.

Для решения вышеперечисленных проблем в [2] был введен т. н. блок пирамидальной субдискретизации (англ. – Pyramid Pooling Module). Его структура приведена на рис. 1.

|  |
| --- |
|  |
| Рисунок 1 – Структура PSPNet. На входе PSPNet принимает карту признаков последнего сверточного слоя (b). Затем применятся слой пирамидальной субдискретизации с фильтрами различного размера (c). После повышения разрешения слои от фильтров разных размеров конкатенируются с исходной картой признаков, а затем производится свертка до итогового предсказания |

Этот блок производит субдискретизацию по фильтрам с различными размерами окна. Это позволяет выделять признаки, относящиеся одновременно и к крупным объектам, определяющим сцену, и к небольшим объектам на ней. Количество различных фильтров, по которым производится субдискретизация, а также их размер выбираются эмпирически. В [2] проведены эксперименты на нескольких различных наборах данных по поиску наиболее оптимальной конфигурации этого блока. Авторы остановились на 4 фильтрах с размерами а в качестве операции для субдискретизации использовалось взятие среднего арифметического.

# 3 Архитектура сети

Сеть строилась с использованием предобученной модели ResNet-101. Выходная карта признаков последнего слоя ResNet передавалась на вход блоку пирамидальной субдискретизации. В нем параллельно на всех фильтрах разного размера проводилась субдискретизация с вычислением среднего, затем свертка до глубины в 512, затем нормализация батчей и, наконец, ReLU-слой (англ. – Rectified Linear Units). Так как на выходе все эти слои конкатенировались с исходной картой признаков (в нашей архитектуре ее глубина 2048), а слоя всего 4, то на выходе блока мы получили карту признаков с глубиной в 4096. Последний слой сети проводит свертку в два этапа , между которыми снова производится нормализация батчей, добавляется нелинейность с помощью ReLU-слоя и удаляются связи между узлами с вероятностью 0.1.

# 4 Экспериментальные результаты

Авторы [2] для обучения на наборе данных cityscapes использовали следующие гиперпараметры: 90 000 итераций, уменьшение веса 0.0001, начальная скорость обучения 0.01. В качестве оптимизатора авторы использовали стохастический градиентный спуск с накоплением Нестерова.

В этой работе обучение проводилось на 10 000 итераций, и в качестве оптимизатора использовался Adam [4]. Остальные параметры не изменялись. Кроме того, для увеличения обучающей выборки использовались случайное масштабирование, случайное горизонтальное симметричное отображение и случайный поворот изображения на угол в интервале как и у авторов [2].

На рисунке 2 представлен график функции потерь на тренировочной выборке.

|  |
| --- |
|  |
| Рисунок 2 – График функции потерь. Скачки у функции образуются на каждой 250-ой итерации, когда производился вынужденный перезапуск процесса обучения |

На рисунке 3 представлен один и тот же кадр валидационной выборки, и результаты предсказаний на нем на нескольких разных этапах.

|  |  |
| --- | --- |
| (a) | |
| (b) | (c) |
| (d) | (e) |
| Рисунок 3 – (a) – Исходное изображение; (b) – размеченное изобрежение; (c) – предсказания после 250 итераций обучения; (d) – предсказания после 4750 итераций; (e) – предсказание после 8750 итераций | |

Также проводилось сравнение результатов классификации предложенного метода и наипростейшего вероятностного классификатора. По всем изображениям из обучающей выборки для каждого пикселя считалась вероятность появления того или иного класса. Затем это полученное статичное изображение-предсказание (рис. 4) также проходило валидацию.

|  |
| --- |
|  |
|  |
| Рисунок 4 – Статичное изображение-предсказание, рассчитанное по вероятностям появления класса попиксельно |

На таблице 1 показаны результаты классификации на тестовой и тренировочной выборке для предложенного метода и вероятностного классификатора. В качестве метрик создатели набора данных использовали точность (англ. – Accuracy): ; и среднее пересечение по объему (англ. – Mean intersection-over-union)[5]: .

Таблица 1 – Метрики точности на валидационной и тренировочной выборке для предложенной и вероятностной классификации

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dataset | Baseline | | PSPNet | |
| Accuracy | Mean IoU | Accuracy | Mean IoU |
| Train | 0.193610 | 0.077194 | 0.632789 | 0.139415 |
| Validation | 0.124558 | 0.056323 | 0.609576 | 0.132633 |

Листинг кода представлен в онлайн репозитории [6].

# Список использованных источников

1. Эталонный тест на наборе данных Cityscapes [Электронный ресурс]. Режим доступа: <https://www.cityscapes-dataset.com/benchmarks/> (дата обращения: 23.12.2017)
2. Zhao H., Shi J., Qi X., и др. Pyramid scene parsing network // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.
3. He K., Zhang X., Ren S., и др. Deep residual learning for image recognition Proceedings of the IEEE conference on computer vision and pattern recognition / 2016. 770–778 с
4. Kingma D., Ba J. Adam: A method for stochastic optimization // 2014
5. Everingham M. et al. The pascal visual object classes challenge: A retrospective //International journal of computer vision. – 2015. – Т. 111. – №. 1. – С. 98-136.
6. Онлайн репозиторий GitHub [Электронный ресурс]. Режим доступа: <https://github.com/ShepelIlya/SegmentationLearning> (дата обращения 25.12.2017)