Санкт-Петербургский Государственный Университет
Математико-механический факультет

Кафедра системного программирования

Спирин Егор Сергеевич

Разработка мобильного приложения для семантической сегментации изображений при построении панорамного изображения

Курсовая работа

Научный руководитель:
ст. преп. Смирнов М.Н.

Санкт-Петербург
2019

SAINT-PETERSBURG STATE UNIVERSITY

Department of System Programming

Spirin Egor

Development of a mobile application for semantic segmentation of images when building a panoramic image

Course Work

Scientific supervisor:
Senior lecturer Smirnov Michail

Saint-Petersburg
2019

Оглавление

Введение

В настоящее время создание качественных фотоснимков занимает одну из приоритетных задач при создание камер и разработке программного обеспечения для мобильных устройств.

Производители смартфонов уже внедряют различные нейросетевые модели для повышения качества сделанной фотографии, но когда речь касается снятия панорамы, то всё ещё остаются проблемы при обработке. Так как при пересечение нескольких фотографий для получения цельной картины необходимо учесть много факторов, например, движущиеся объекты или световой баланс.

Анализ подвижных объектов напрямую связан с семантической сегментацией каждой фотографии. При этом сама задача является хорошо изученной, имеется много работ на эту тему, но далеко не каждый подход удовлетворяет требованиям мобильных приложений.

Исходя из этого, в рамках данной работы, будет разработана модель, способная сохранять качество сегментации в условиях ограниченного времени и маленьких ресурсов.

Рис. 1: Пример неправильной склейки панорамы

1. Цель работы

Цель данной работы – это поиск оптимального метода семантической сегментации фотографии, чтобы в дальнейшем внедрять его в приложение для склейки панорамного изображения. Под оптимальным понимается удовлетворение ключевых характеристик, а именно точность сегментация, время работы и занимаемое место на устройстве.

Для достижения данной цели, поставленны следующие задачи:

1. Сбор и подготовка данных;
2. Изучение методов семантической сегментации изображений;
3. Разработка android-приложения, для сравнения подходов к сегментации по всем необходимым метрикам на различных устройствах;
4. Внедрение изученных методов обработки изображений в приложение;
5. Сбор и анализ результатов работы всех подходов на валидационном множестве.

2. Обзор

2.1. Семантическая сегментация

Семантическая сегментация – это задача присвоения метки класса каждому пикселю на изображению. Таким образом показывается ”понимание” изображения, не только, что находится на нём, но и где.

Рис. 2: Пример семантической сегментации (исходное изображение, маска сегментации, наложение маски на исходное изображение)

До наступления эпохи глубокого обучения для сегментации применялись самые разнообразные техники обработки изображений в зависимости от области интересов. Например, метод пороговой сегментации [11], выделение границ объектов на основе градиентов [12] или с помощью выделения сообществ на графе [13].

С ростом популярности нейронных сетей, все эти методы были вытеснены на задний план, так как их качество заметно ниже, поэтому в данной работе будут исследоваться только методы сегментации с использованием нейронных сетей.

2.2. Оценка качества

Так как конечная модель будет внедряться в приложение для работы с пользователями, то накладываются различные ограничения на её качество, время работы и размер в памяти.

Для сравнения различных подходов, из общих данных выделяется специальное тестирующее множество, на котором будет запускаться

каждая модель и полученные результаты будут агрегироваться в общую таблицу для дальнейшего анализа.

2.2.1. Время работы

Время работы каждой модели считается, как среднее время работы по всем изображениям из тестирующего набора.

$$mT = \frac{1}{n}\sum_{i=1}^{n} t_i$$

где:

- n – количество изображений в выборке;
- t_i – время работы сегментации i изображения.

2.2.2. Качество сегментации

Для оценки качества сегментации применяется стандартная метрика в задачах обработки изображения – *intersection over union*, которая оценивает близость предсказанной маски к оригинальной. Для каждого класса объектов, за I обозначается количество пикселей, где класс предсказанной маски совпадает с оригинальным, а за U количество пикселей, где хотя бы одна из масок указала на объект данного класса. Тогда качество нахождения данного класса

$$IoU_{class} = \frac{I}{U}$$

А качество сегментации изображения – среднее значение данной метрики по всем классам

$$IoU = \frac{1}{C}\sum_{i=1}^{C} IoU_{class_i}$$

где:

- C – количество классов объектов;

- IoU_{class_i} – качество сегментации класса i.

Качество модели оценивается, как среднее по всем изображениям тестирующего множества

$$mIoU = \frac{1}{n}\sum_{i=1}^{n} IoU_i$$

2.3. Данные

В работе используется датасет Pascal VOC [6], в нём представленные изображения и маски к ним, рассчитанные на 20 классов, которые часто встречаются на фотографиях и необходимы для улучшения качества склеивания панорамных снимков: люди, автомобили, животные (кошки, собаки) и т.д.

Датасет включает в себя 1464 тренировочных изображений и 1449 валидационных изображений, которые были объединены в общий датасет и разбиты на тренировочный, валидационный и тестирующий наборы данных в соотношение $7 : 2 : 1$. Обучение происходило на тренировочном наборе данных с постоянным контролем значение функции потерь на валидационном множестве. Все конечные оценки замерялись на тестирующем множестве.

Так же в процессе обучения данные подвергаются процессу аугментации, чтобы получить больше различных изображений и лучше натренировать нейронную сеть.

3. Методы глубокого обучения

В задачах обработки изображений в основном используются свёрточные сети архитектуры кодер-декодер. Согласно [10], где можно сравнивать различные архитектуры работающие с одинаковыми данными, такой подход занимает лидирующие положения в задачах сегментации, детекции, классификации и т.п. В данном разделе будут рассмотрены архитектуры нейронных сетей приспособленных к работе на мобильных устройствах.

3.1. U-Net

Нейронная сеть, которая пришла из области обработки медицинских снимков [8], на данный момент является одной из самых популярных.

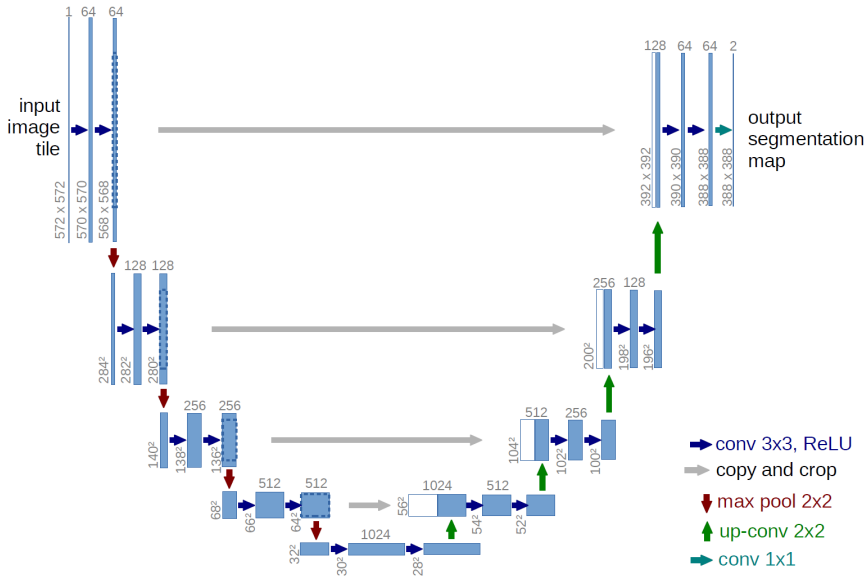

Рис. 3: U-Net архитектура (пример для размера 32×32 на выходе кодера). Голубым обозначен мультиканальные карты признаков, количество каналов обозначено над слоем. Белым обозначен скопированные карты признаков. Стрелки указывают на различные операции. С левой стороны каждого слоя указан $x - y$ размер входа.

Её отличительная особенность – это хороший результат даже при использование малого количества данных, при этом использование различных способов кодирования сохраняет высокое качество сегментации [7].

В качестве кодера для мобильного устройства существует специальная сеть MobileNetV2 [4], которая обладает маленьким размером, что является преймуществом при портирование на мобильные устройства.

Таким образом в данной работе будет рассмотрена сегментация с помощью MobileNetV2 + U-Net.

3.2. DeepLabV3

Данная архитектура является лидером в задаче семантической сегментации [10], её отличительной особенностью стали использование расширенных свёрточных слоёв и объединение пирамидальной субдискретизации [2].

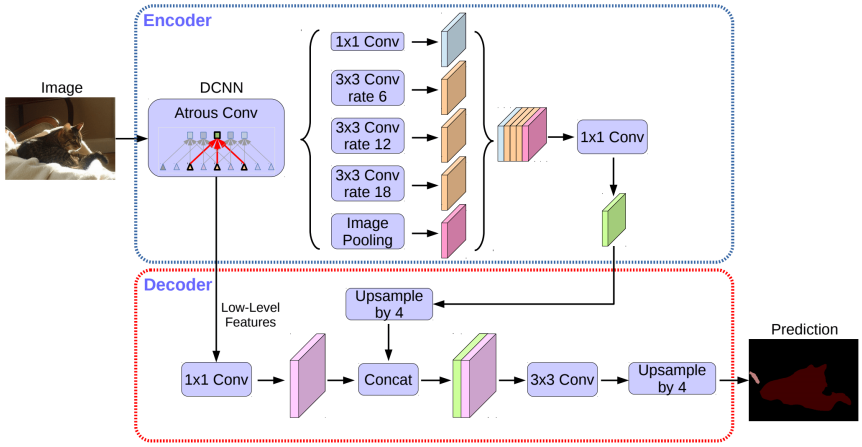

Рис. 4: DeepLabV3 архитектура. Кодер использует многомасштабную контекстную информацию с помощью применения свёрток к изображению разного масштаба, а достаточно простой декодер уточняет границы сегментации.

Данный подход позволяет также внедрить MobileNetV2 [4] в часть кодера, что делает эту нейронной сетью крайне быстрой и качественной.

В ходе работы рассмотрены 2 нейронной сети этой архитектуры:

- DeepLabV3 GPU – сегментация изображения размером 251×251 с использованием GPU устройства;
- DeepLabV3 CPU – сегментация изображения размером 513×513.

Первый подход направлен на ускорение времени работы, а второй на высокое качество сегментации.

3.3. E-Net

Авторы архитектуры E-Net [1] решают задачу сегментации в режиме реального времени [1], что подразумевает максимальную скорость обработки каждого изображения с минимальными потерями в качестве.

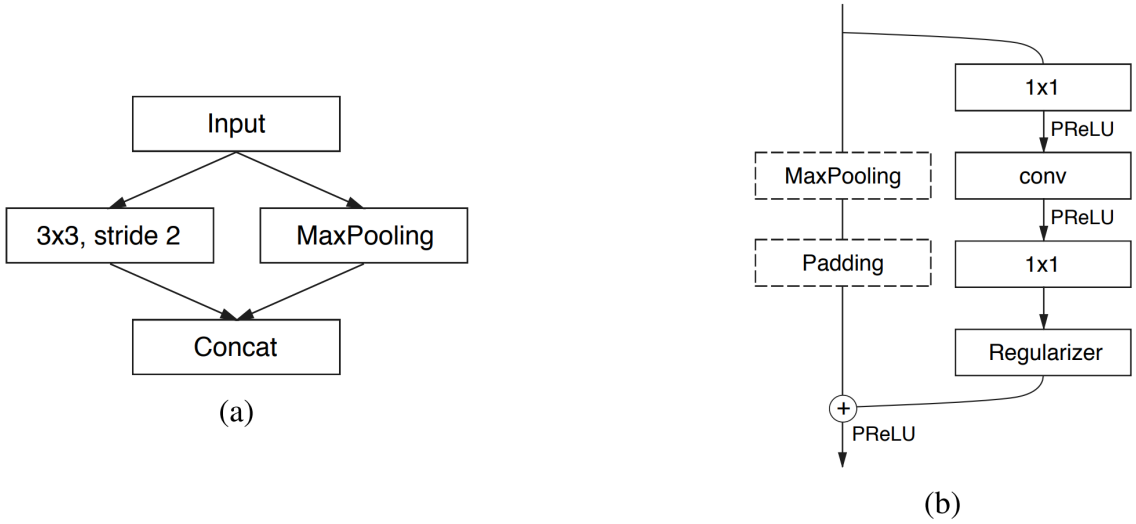

Рис. 5: Блоки E-Net архитектуры. (a) Начальный блок, субдискретизация функцией максимума и окном 2×2 и свёртка размером 3×3 с 13 фильтрами на выходе. (b) Bottleneck модуль, при этом свёрточный слой может быть как в обычном понимание, так и расширенной свёрткой или разворачивающей.

Данная модель состоит из bottleneck модулей, где на разных этапах применяются разные типы свёрток. Благодаря значительному уменьшению числа параметров и FLOPS время работы данной сети существенно уменьшилось.

3.4. IC-Net

IC-Net [3] предназначена для решения задачи сегментации в режиме реального времени. Такая нейронная сеть извлекает признаки из одного изображения приведённого к разным размерам, а затем объединяет в общую маску.

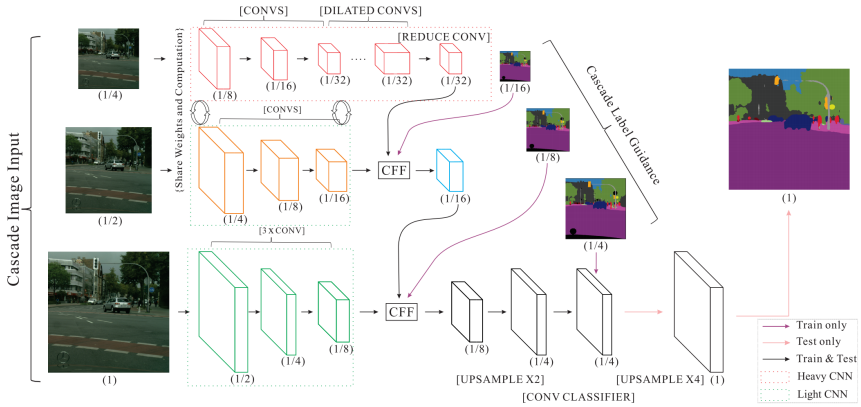

Рис. 6: IC-Net архитектура. *CFF* обозначает каскадное объединение признаков.

4. Мобильное приложение

Разработка android-приложения является необходимым условием для сравнения подходов к решению задачи семантической сегментации, так как необходимо получить информацию о работе всех методов с устройств с различными характеристиками.

В качестве основного языка программирования выбран язык *Kotlin*, так как сейчас это является стандартом де-факто при разработке android-приложения. Основным фреймворком для запуска нейронных сетей на устройстве выбраны *TensorFlow Lite* [9], OpenSource библиотека позволяющая запускать нейронные сети на мобильных устройствах с использованием GPU, но не покрывающая весь необходимый функционал, и *TensorFlow Mobile*, предыдущая версия библиотеки для запуска моделей на устройстве, покрывает весь функционал, но уступает в производительности. В качестве альтернативы был рассмотрен фреймворк *OpenCV Android* [5], но он требует установки дополнительного приложения на устройство, либо в разы увеличивает размер приложения, поэтому решено было откзаться от его использования.

Конечное приложение поддерживает такие функции, как

- выбор изображения для сегментации из галереи, тестирующего множества или фотографии с камеры;
- выбор нейронной сети для сегментации;

- сегментация изображения, отображение результатов, затраченного времени и точности, для изображений из тестирующей выборки;

- сбор информации о работе всех установленных методов сегментации, включает в себя запуск каждой нейронной сети на тестирующих изображениях, сбор статистит и выгрузка на сервер.

Исходный код можно найти в репозитории `https://github.com/SpirinEgor/mobile_semantic_segmentation`.

5. Результаты

В данном разделе представлены результаты сравнения обученных нейронных сетей по основным критериям. Оценка качества проводилась на тестирующем множестве, независимо от устройства. Размер модели – это замороженная модель, сконвентированная под мобильное устройство. Среднее время работы оценивалось на различных устройствах на одном и том же тестирующем наборе данных без учёта пре- и постпроцессинга.

Нейронная сеть	*mIoU*	Размер (mb)
DeepLabV3 GPU	0.631	2.8
DeepLabV3 CPU	0.797	8.8
U-Net	0.612	25.4
E-Net	0.307	1.7
IC-Net	0.639	27.1

Таблица 1: Оценка качества и занимаемый в памяти размер

Модель	Время работы (ms)		
	Snap. 625 (4)	Snap. 845 (6)	Snap. 660 (4)
DeepLabV3 GPU	391.6	139.55	209.1
DeepLabV3 CPU	2307.75	845.35	1464.3
U-Net	867.35	414.68	557.55
E-Net	307.6	106.95	175.85
IC-Net	375.05	127.6	199.4
	Snap. 808 (3)	Snap. 430 (3)	Exynos 7420 (3)
DeepLabV3 GPU	325.3	630.85	180.45
DeepLabV3 CPU	3525.45	5177.05	2328.35
U-Net	1426.05	1933.13	975.0
E-Net	430.9	590.76	209.05
IC-Net	531.15	656.4	248.85

Таблица 2: Время работы моделей на различных устройствах. *Snap.* – Snapdragon. В скобках размер оперативной памяти в гигабайтах.

Таким образом, модель *DeepLabV3 CPU* обладает самым высоким качеством, но при этом обрабатывает фотографию дольше всего, модели *IC-Net* и *E-Net*, как и ожидалось от моделей, работающие в real-time,

показывают высокую скорость, однако уступают в качестве. Модель *DeepLabV3 GPU* смогла приблизится к ним, а на некоторых устройствах даже обогнать, сохранив при этом хорошее качество. *U-Net* показывает среднее время работы и низкое качество, это вызвано тем, что она предсказывает появление многих классов, которых на самом деле нет, хотя основные классы находятся хорошо.

Заключение

В результате работы были выполнены следующие задачи:

- изучены подходы к решению задачи семантической сегментации;
- рассмотрены архитектуры нейронных сетей для сегментации на мобильных устройствах;
- сбор и подготовка данных для обучения;
- разработка мобильного приложения для запуска и тестирования нейронных сетей;
- обучение и внедрение в мобильное приложение моделей для семантической сегментации изображений;
- сбор и анализ результатов работы всех реализованных методов.

На этом исследование не заканчивается, так как существуют направления, в которых можно продолжать работу, например, изучение и внедрение других архитектур: *PSP-Net*, *FCN*, *MASK-RCNN*...

Так же можно продолжить работать с уже реализованными методами, так как данные модели можно ещё дообучать на новых данных и оптимизировать под устройства для уменьшения времени работы.

Список литературы

[1] ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation / Adam Paszke, Abhishek Chaurasia, Sangpil Kim, Eugenio Culurciello // CoRR. — 2016. — Vol. abs/1606.02147. — 1606.02147.

[2] Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation / Liang-Chieh Chen, Yukun Zhu, George Papandreou et al. // CoRR. — 2018. — Vol. abs/1802.02611. — 1802.02611.

[3] ICNet for Real-Time Semantic Segmentation on High-Resolution Images / Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen et al. // CoRR. — 2017. — Vol. abs/1704.08545. — 1704.08545.

[4] Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation / Mark Sandler, Andrew G. Howard, Menglong Zhu et al. // CoRR. — 2018. — Vol. abs/1801.04381. — 1801.04381.

[5] OpenCV. — 2019. — Access mode: `https://opencv.org/platforms` (online; accessed: 13.05.2019).

[6] The Pascal Visual Object Classes Challenge: A Retrospective / M. Everingham, S. M. A. Eslami, L. Van Gool et al. // International Journal of Computer Vision. — 2015. — Jan. — Vol. 111, no. 1. — P. 98–136.

[7] RTSeg: Real-time Semantic Segmentation Comparative Study / Mennatullah Siam, Mostafa Gamal, Moemen Abdel-Razek et al. // CoRR. — 2018. — Vol. abs/1803.02758. — 1803.02758.

[8] Ronneberger Olaf, Fischer Philipp, Brox Thomas. U-Net: Convolutional Networks for Biomedical Image Segmentation // CoRR. — 2015. — Vol. abs/1505.04597. — 1505.04597.

[9] TensorFlow Lite. — 2019. — Access mode: https://www.tensorflow.org/lite (online; accessed: 13.05.2019).

[10] Visual Object Classes Challenge 2012. — 2012. — Access mode: http://host.robots.ox.ac.uk/pascal/VOC/voc2012/ (online; accessed: 13.05.2019).

[11] А. Ю. Тропченко А.А. Тропченко. Методы вторичной обработки и распознования изображений / Под ред. Университет ИТМО. — Санкт-Петербург : Учебное пособие, 2015. — С. 17.

[12] А. Ю. Тропченко А.А. Тропченко. Методы вторичной обработки и распознования изображений / Под ред. Университет ИТМО. — Санкт-Петербург : Учебное пособие, 2015. — С. 44.

[13] С.В. Белим С.Б. Ларионов. Сегментация изображений на основе алгоритма выделения сообществ на графе // Математические структуры и моделирование. — 2016. — Vol. 3(39). — P. 74–85. — Access mode: http://msm.omsu.ru/jrns/jrn39/BelimLarionovSB.pdf.