АНАЛИЗ СТРУКТУР НЕЙРОННЫХ СЕТЕЙ ENCODER-DECODER ДЛЯ СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЙ НА МОБИЛЬНЫХ УСТРОЙСТВАХ

Бычик Юрий Григорьевич

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

Ярмолик Вячеслав Николаевич – доктор технических наук

В данном докладе рассмотрены архитектуры нейронных сетей UNet и LinkNet класса Encoder-Decoder, позволяющие использовать сверточные сети для классификации изображений для решения задачи сегментации. Выполнена оценка применимости нейронных сетей вида Encoder-Decoder для запуска на мобильных устройствах. Описана нейронная сеть MobileNetV2 для классификации и способ ее использования в архитектурах UNet и LinkNet. Отражены экспериментальные результаты по оценке скорости работы нейронных сетей на мобильном устройстве.

Наиболее популярным классом нейросетевых методов для решения задач компьютерного зрения являются сверточные сети. Оператор свертки является основным вычислительным блоком. Последовательное применение свертки с точки зрения нейронных сетей позволяет выделять признаки “низкого уровня” (линии, точки, и т.п.) и далее аккумулировать признаки “низкого уровня” в признаки “высокого уровня” (контуры объектов и т.п.). В цифровом представлении найденные признаки представляют собой многомерные тензоры и используются для классификации изображений [1].

Аккумулированные признаки можно “трансформировать” в сегментированные области на исходном изображении с помощью некоторого обратного преобразования. Использование нейронных сетей для классификации в качестве основы для сегментации представляет собой архитектуру класса Encoder-Decoder. Наиболее часто используемыми архитектурами Encoder-Decoder являются UNet и LinkNet. Для работы UNet и LinkNet на мобильных устройствах необходимо использовать адаптированную для мобильных устройств сеть для классификациии MobileNetV2 [2].

Скорость работы архитектур Encoder-Decoder линейно зависит от скорости работы сети для классификации MobileNetV2, которая используется в качестве Encoder-части. Коэффициент линейности зависит от скорости обратных преобразований в Decoder-части, как правило от количества сверток. Таким образом, нейронные сети типа Encoder-Decoder для сегментации на основе MobileNetV2 применимы для мобильных устройств [3].

MobileNetV2 состоит из начального слоя с 32 сверточными фильтрами и 17 остаточных блоков (residual blocks, bottlenecks). Каждый блок выполняет “распаковку-сжатие” признаков. На вход блока поступают трехмерные тензоры с N каналами. Далее идет оператор распаковки (expansion), где с помощью набора 6N сверточных фильтров размерностью 1x1xN количество каналов увеличивается до 6N, и на выходе каждой свертки дополнительно используется нелинейная функция Relu6. Далее применяются 6N сверточных фильтров размерностью 3x3x1 с разделением по глубине (depthwise convolution). К результату каждого фильтра применяется нелинейная функция Relu6. На последнем этапе происходит “сжатие” количества каналов тензора с помощью M сверток размерности 1x1x6N до размерности глубины M > N, где M ~ в 1,5 - 2 раза больше N. Для некоторых остаточных блоков M равно N для совпадения входных и выходных тензоров. Это позволяет совместить результаты работы остаточного блока с входными данными для лучшего обучения сети. На предпоследнем этапе получается тензор малой размерности с большим количеством каналов D, который сжимается последним оператором до размерности 1x1xD. Далее используется полносвязный слой с D входами и R выходами, которые представляют собой вероятности искомых классов. Описание операторов MobileNetV2 представлено в таблице 1.

Таблица 1 – Операторы MobileNetV2

Размер тензора, вход	Оператор	Кол-во каналов, выход	Кол-во операторов
224x224x3	Conv 2D 3x3	32	1
112x112x32	Bottleneck	16	1
112x112x16	Bottleneck	24	2
56x56x24	Bottleneck	32	3
28x28x32	Bottleneck	64	4
14x14x64	Bottleneck	96	3
14x14x96	Bottleneck	160	3
7x7x160	Bottleneck	320	1
7x7x320	Conv 2D 1x1	1280	1
7x7x1280	Average pooling, 7x7	1280	1
1x1x1280	Conv2d 1x1	R	1

Для обработки входного изображения размером 224x224x3 выполняется ~600 миллионов операций с плавающей точкой (FLOPS), нейронная сеть содержит ~3.5 миллионов параметров.

В таком виде у сегментации есть проблемы с определением границ. Это связано с тем, что признаки “высокого” уровня, полученные в результаты работы сети, не учитывают локальный контекст каждого признака, т.е детали связи некоторой области изображения с соседними областями. Для решения этой задачи используются архитектуры UNet[4], LinkNet[5], которые на каждом этапе увеличения высоты и ширины тензора добавляют к данному тензору каналы из этапов Encoder-части. В UNet архитектуре используется операция concatenate, которая увеличивает общее количество каналов на каждом этапе работы Decoder-части. В архитектуре LinkNet используется операция add, которая складывает значения соответствующих каналов в связанных тензорах Encoder- и Decoder-частях. Схемы UNet и LinkNet приведены на рисунке 1.

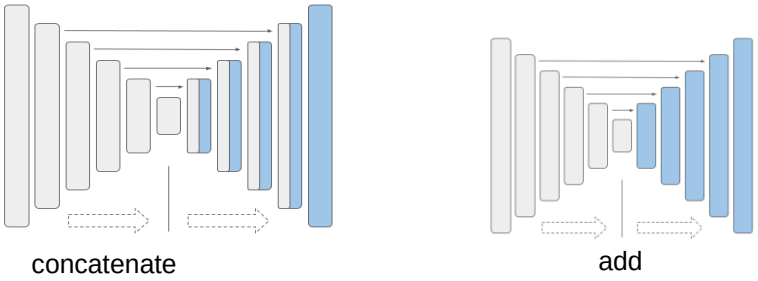

Рисунок 1 – Архитектуры Encoder-Decoder: UNet (слева), LinkNet(справа)

Для оценки скорости работы архитектур UNet и LinkNet с использованием MobileNetV2 разработано программное средство на языке Python с использованием фреймворка TensorFlow 2 [6]. С помощью данных инструментов выполнялось моделирование архитектур сетей UNet и LinkNet. Для тренировки исходной нейронной сети MobileNetV2 был использован набор данных ImageNet. Для тестирования скорости работы сетей на мобильных устройствах использовалась платформа Android, фреймворк TensorFlow Lite, 8-ядерный процессор Snapdragon 845 с частотой 2GHz. Экспериментальные результаты представлены в таблице 2.

Таблица 2 – Экспериментальная оценка скорости сегментации 1 изобр. с помощью UNet и LinkNet

Нейронная сеть	Скорость сегментации 1 изобр., мс.	Отношение к MobileNetV2
MobileNetV2	~70	1
UNet + MobileNetV2	~350	~5
LinkNet + MobileNetV2	~150	~2

Скорость работы сети для сегментации изображений больше, чем классификации. Однако данные архитектуры применимы для мобильных устройствх, т.к. абсолютные значения скорости работы приемлемыми для решения классов прикладных задач, не связанных с реальным временем.

Список использованных источников:

1. Гонсалес, Р. Цифровая обработка изображений, издание 3-е, исправленное и дополненное / Гонсалес Р., Вудс Р. – Изд. 3-е. – М. : Техносфера, 2012. – 1104 с.

2. Goodfellow I. Deep Learning / Goodfellow I., Bengio J., Courville A. – MIT Press, 2016 – 800 p.

3. Sandler M. MobileNetV2: Inverted Residuals and Linear Bottlenecks / Sandler M., Howard A.G., Zhu M. - 2018 IEEE / CVF Conference on Computer Vision and Pattern Recognition, 2018. – P. 4510-4520

4. Ronneberger O. U-Net: Convolutional Networks for Biomedical Image Segmentation / Ronneberger O, Fischer P., Brox T. - CoRR, vol. abs/1505.04597 – 2015

5. Chaurasia A. LinkNet: Exploiting encoder representations for efficient semantic segmentation / Chaurasia A, Culurciello E. - 2017 IEEE Visual Communications and Image Processing (VCIP), 2017 – P.1-4

6. Gulli A. Deep Learning with TensorFlow 2 and Keras: Regression, ConvNets, GANs, RNNs, NLP, and more with TensorFlow 2 and the Keras API / Guill A., Kapoor A. - Packt Publishing, 2019 – 646 p.