Санкт-Петербургский государственный университет

АЛИЕВ Фарамоз Серверович

Выпускная квалификационная работа

Восстановление трехмерной сцены по набору изображений

Уровень образования: бакалавриат

Направление 01.03.02 «Прикладная математика и информатика»

Основная образовательная программа СВ.5005.2016 «Прикладная математика, фундаментальная информатика и программирование»

Профиль: «Исследование и проектирование систем управления и обработки сигналов»

Научный руководитель:
заведующий кафедрой компьютерных технологий и систем,
доктор физ.-мат. наук,
профессор Веремей Е.И.

Рецензент:
ассистент кафедры теории систем управления электрофизической аппаратурой,
Пономарев В.А.

Санкт-Петербург
2020

Содержание

Введение

В последние десятилетия наблюдается существенный прогресс в построении трехмерной модели сцены по набору изображений или видеоряду. Эти работы стали востребованы в связи с необходимостью наполнения систем виртуальной реальности данными из реальных сцен.

Так, например, при создании фильмов все чаще прибегают к 3D-моделированию. С помощью камер и датчиков воссоздается модель сцены и накладываются на нее эффекты, что гораздо быстрее, чем создание модели с нуля. Однако область применения 3D-реконструкции обширна и не ограничивается кинематографом. Сюда входит робототехника, археология, медицина, машиностроение, архитектура, дизайн. Также все чаще при создании анимации в компьютерных играх прибегают к 3D-моделированию реальных сцен и людей.

Все это делает задачу восстановления трехмерной модели по набору изображений одной из основных в области компьютерного зрения. На сегодняшний день представлено множество различных решений. В простейшем случае, модель может представлять из себя набор точек трехмерного пространства. Более же сложные методы строят полную трехмерную модель.

Существует два класса методов для решения данной задачи. Активные методы используют вспомогательные оборудования: различные 3D-сенсоры, -датчики, -сканеры, лазерные дальномеры. Все эти приспособления стоят очень дорого и могут быть применены не везде, но дают точный результат. Пассивные же методы не требуют больших материальных затрат и основаны на обработке изображений, полученных с одной или нескольких камер.

Пассивные методы различают по ограничениям, которые накладываются на входные данные. Это может быть стереопара изображений, видеоряд с движущейся в пространстве камерой или, наоборот, со статической камерой, но

обязательно движущимся объектом. Рассмотрим пассивный метод построения трехмерных моделей по одному изображению с применением нейронных сетей.

Один из самых простых способов решения данной задачи подразумевает использование уже готовых 3D-моделей различных объектов, которые могут встречаться на фотографиях сцен. Главная задача состоит в том, чтобы определить какие именно объекты изображены на фотографии, и как они расположены в пространстве.

Таким образом, основными этапами метода являются:

- Семантическая сегментация.
- Построение карты глубин.

Семантическая сегментация изображения — это разделение изображения на отдельные группы пикселей, области, соответствующие одному объекту с одновременным определением типа объекта в каждой области.

Карта глубины — это изображение, на котором для каждого пикселя, вместо цвета, хранится его расстояние до камеры.

Особенное внимание уделим капсульным нейронным сетям, так как эта архитектура появилась совсем недавно и количество решений различных задач с их применением мало, а результаты превосходят многие известные методы.

Обзор литературы

Рассмотрим задачу семантической сегментации изображений. Представленная в 2017 году на конференции модель сверточной нейронной сети [1] впервые смогла превысить среднюю точность сегментации 80%. Её модификация [2] имеет схожий результат, но обучается в два раза быстрее и использует при это в два раза меньше видеопамяти. Другая более сложная модель [3], представленная год назад, имеет самую высокую точность в 82%, но для её обучения требуются большие мощности.

В другой статье [4] описана модель SegCaps, основанная уже на капсульных нейронных сетях. Она была применена к задаче семантической сегментации медицинских изображений. По этой причине её сравнивают только с другой моделью [5], основанной на сверточных нейронных сетях и предназначенной для той же цели. Как сказано в работе, результаты капсульной сети выше, чем у сверточной.

Для построения карт глубин существует большое количество различных моделей. Некоторые из них представлены в статьях [6], [7], [8]. Для сравнения большого количества моделей был создан специальный бенчмарк (The KITTI Vision Benchmark), содержащий 93 000 фотографий различных сцен из разных датасетов [9]. Все эти модели основаны на сверточных нейронных сетях.

В работе [10] рассматривается метод SLAM (Simultaneous Localization And Mapping) с использованием карт глубин, построенных при помощи капсульной нейронной сети. В статье используется baseline-модель CapsNet [11]. Из результатов видно, что их метод дает более точные результаты, чем другой, использующий сверточные сети.

Постановка задачи

В данной работе будет рассмотрена задача предсказания карты глубины изображения с помощью капсульной нейронной сети SegCaps [4], предназначенная для сегментации изображений.

На вход алгоритму подается цветное изображение в виде матрицы *I* размером NxMx3, элементами которой являются целые числа от 0 до 255. Необходимо построить карту глубин для данного изображения в виде матрицы *D* размером NxMx1, элементами которой будут вещественные числа от 0 до 10 (расстояние от камеры, указанное в метрах).

Для обучения будем использовать датасет NYU Depth Dataset V2 [12], состоящий из 1449 RGB-D изображений (рис. 1).

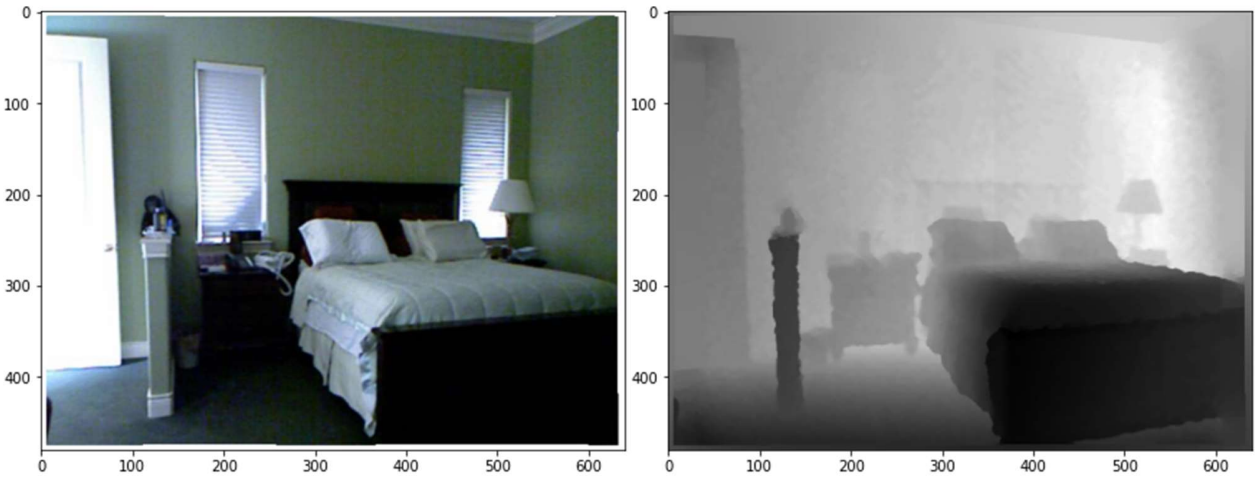

Рис. 1 Пример изображения и карты глубины.

Глава 1. Архитектуры нейронных сетей для построения карт глубин

1.1. Сверточные нейронные сети

Сверточная нейронная сеть (англ. convolutional neural network, CNN) — это архитектура искусственных нейронных сетей, нацеленная на эффективное распознавание образов. В ней используются некоторые особенности зрительной коры, в которой были найдены простые клетки, реагирующие на прямые линии под разными углами, и сложные клетки, реакция которых связана с активацией определённого набора простых клеток. Таким образом, эта сеть представляет из себя чередование сверточных слоев и слоев субдискретизации.

Сверточный слой представляет из себя применение операции свертки к выходам предыдущего слоя, где веса ядра свертки являются обучаемыми параметрами. Еще один обучаемый вес — это константный сдвиг. В одном таком слое может быть несколько сверток, причем свертки могут быть трехмерными (для цветных изображений, рис. 2).

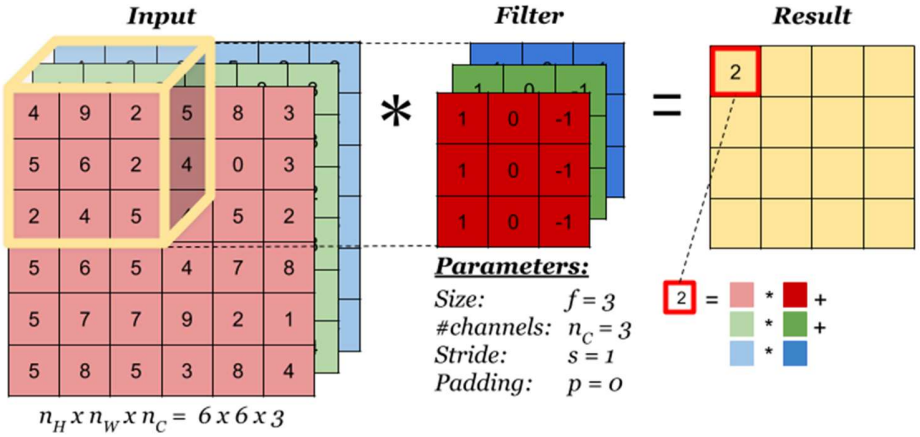

Рис. 2 Пример свертки с трехмерным ядром.

Скалярный результат свертки не сразу подается на слой субдискретизации, а проходит через функцию активации, которая представляет собой нелинейную функцию. Самая простая и часто используемая функция активации — ReLU

(англ. rectified linear unit) вычисляется по формуле $f(x) = \max(0, x)$. Её простота позволяет существенно ускорить процесс обучения сети.

Слой субдискретизации призван уменьшать размерность изображения. Для этого исходное изображение делится на блоки (обычно размером 2x2) и для каждого блока вычисляется некоторая функция (рис. 3). Чаще всего используется функция максимума или взвешенного среднего. Обучаемых параметров у этого слоя нет. Он предназначен для ускорения вычислений и уменьшения размерности карт предыдущего слоя. При свертке выявляются некоторые признаки, и дальнейшая обработка подробного изображения не нужна, поэтому изображение уплотняется.

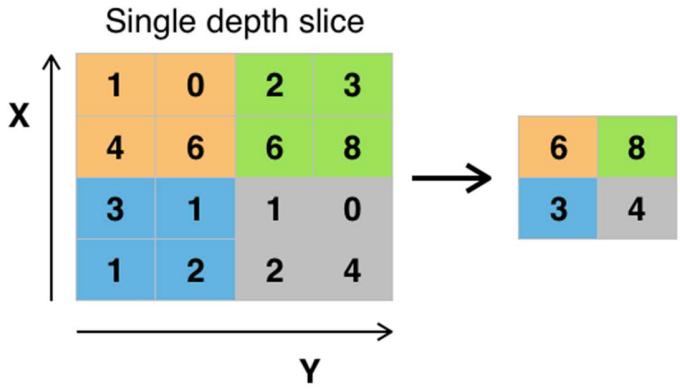

Рис. 3 Субдискретизация с фильтром 2x2 и функцией максимума.

Можно выделить следующие преимущества сверточной нейронной сети:

- Это один из лучших алгоритмов для распознавания и классификации изображений.
- Удобное распараллеливание вычислений, в том числе с использованием графических процессоров.
- Устойчивость к поворотам и сдвигам распознаваемого изображения.
- Гораздо меньшее количество настраиваемых параметров по сравнению с полносвязной нейронной сетью.

Недостатком же является большое количество варьируемых параметров. Количество слоёв, размерность ядра свёртки и их количество на каждом слое, шаг сдвига ядра при обработке слоя, степень уменьшения размерности при субдискретизации, функция по уменьшению размерности (выбор максимума, среднего) – все эти параметры существенно влияют на результат и выбираются исследователями эмпирически.

1.2. Капсульные нейронные сети

В статье [11] впервые было представлено описание капсульной нейронной сети, а также был предложен алгоритм динамической маршрутизации для обучения данной архитектуры.

У классических сверточных сетей существует еще один серьезный недостаток. Внутреннее представление данных не учитывает пространственную иерархию между простыми и сложными объектами (рис. 4). Например, если на изображении в случайном порядке расположить глаза, нос и рот, то для сверточной нейронной сети это все еще будет считаться признаками наличия лица. Также поворот объекта в пространстве существенно уменьшает вероятность распознавания объекта. Капсульные нейронные сети лишены этого недостатка, к тому же ошибка распознавания объекта в другом ракурсе снижена на 45%.

Модели капсульных сетей начинаются с традиционных сверточных слоев. Далее выходы (наборы слоев) разделяются на несколько частей и для каждой части выделяются вектора (капсулы, рис. 5). Капсулы хранят в себе информацию в векторной форме. Длина вектора кодирует вероятность обнаружения объекта, а направление — его состояние (рис. 6).

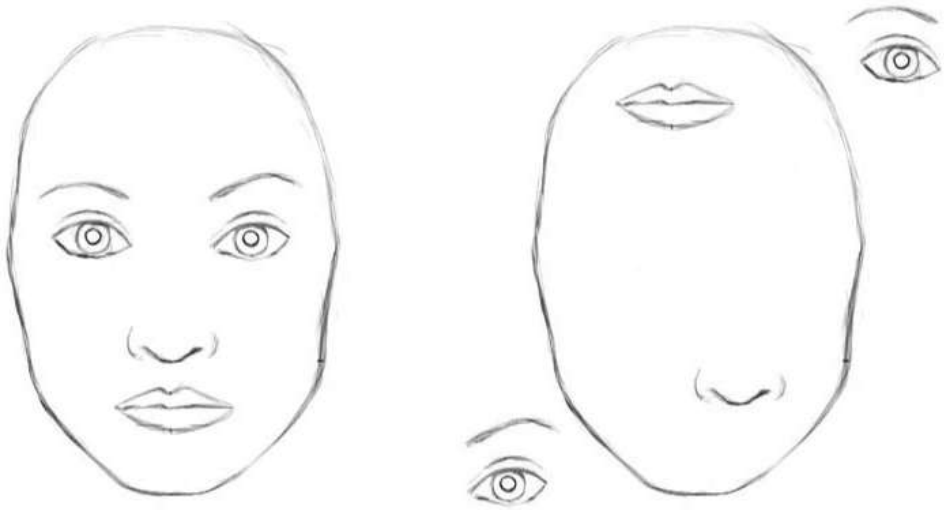

Рис. 4 Пример ошибочного распознавания. Для сверточной нейронной сети оба изображения схожи.

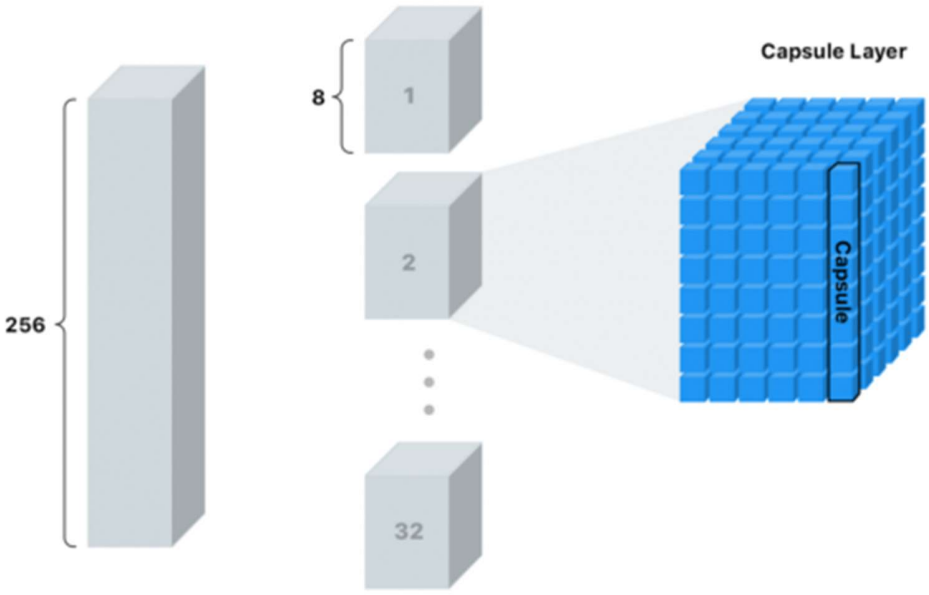

Рис. 5 Капсульный слой. На вход подается выход сверточного слоя размером 256x6x6. Затем все делится на 32 части. В итоге в каждом части находится 36 капсул размерностью 8.

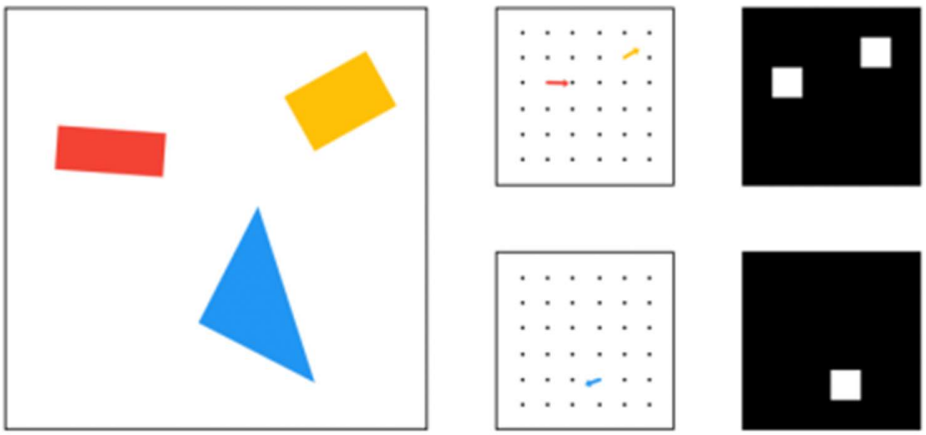

Рис. 6 Упрощенный пример сравнения выходов капсульных слоев. Один для прямоугольников, другой для треугольников. Направление вектора указывает на угол поворота фигуры.

Для связи выхода одного капсульного слоя с другим был предложен алгоритм динамической маршрутизации, представленный на рисунках 7 и 8. На вход алгоритму подаются капсулы нижнего уровня l и их выходы $\hat{u}$, а также количество итераций маршрутизации r. Вначале обнуляется временная переменная. Далее шаги повторяются r раз. Вычисляются веса маршрутизации для каждой капсулы нижнего уровня. Затем следует расчет линейной комбинации выходов капсул нижнего уровня, взвешенных с коэффициентами, рассчитанными на предыдущем шаге. И наконец вектора проходят через нелинейное преобразование (формула представлена на рис. 7), которое гарантирует сохранение направления вектора, но его длина при этом не превышает единицу. Этот шаг создает выходной вектор для капсул более высокого уровня. Оптимальное количество итераций для данного алгоритма является 3.

Capsule		
Input from low-level capsule		$\text{vector}(\mathbf{u}_i)$
Operation	Affine Transform	$\hat{\mathbf{u}}_{j\|i} = \mathbf{W}_{ij}\mathbf{u}_i$
	Weighting	$\mathbf{s}_j = \sum_i c_{ij}\hat{\mathbf{u}}_{j\|i}$
	Sum	
	Nonlinear Activation	$\mathbf{v}_j = \frac{\|\mathbf{s}_j\|^2}{1+\|\mathbf{s}_j\|^2}\frac{\mathbf{s}_j}{\|\mathbf{s}_j\|}$
Output		$\text{vector}(\mathbf{v}_j)$

Рис. 7 Динамическая маршрутизация.

Procedure 1 Routing algorithm.

1: **procedure** ROUTING($\hat{\boldsymbol{u}}_{j|i}, r, l$)
2: for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow 0$.
3: **for** r iterations **do**
4: for all capsule i in layer l: $\mathbf{c}_i \leftarrow \texttt{softmax}(\mathbf{b}_i)$
5: for all capsule j in layer $(l+1)$: $\mathbf{s}_j \leftarrow \sum_i c_{ij}\hat{\mathbf{u}}_{j|i}$
6: for all capsule j in layer $(l+1)$: $\mathbf{v}_j \leftarrow \texttt{squash}(\mathbf{s}_j)$
7: for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow b_{ij} + \hat{\mathbf{u}}_{j|i}.\mathbf{v}_j$
return $\mathbf{v}_j$

Рис. 8 Алгоритм динамической маршрутизации.

Благодаря тому, что в капсулах находится вся дополнительная информация об объекте, можно запустить обратный процесс – реконструкцию изображения по капсулам. На этом этапе блок реконструкции пытается воссоздать исходное изображение только при помощи капсул (рис. 9). Полученное изображение сравнивается с исходным и происходит оценка модели. Все это позволяет более эффективно обучать модель.

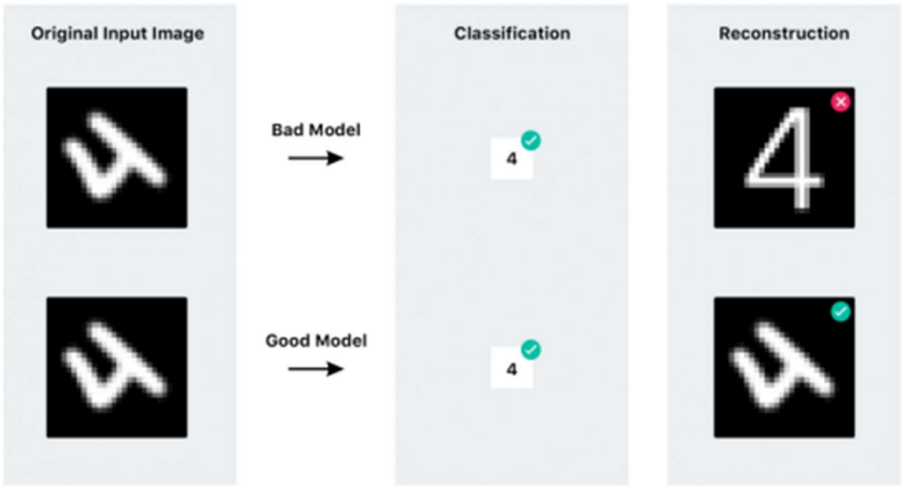

Рис. 9 Пример хорошей и плохой модели.

Стоит отметить, что количество обучаемых параметров у капсульных сетей значительно меньше, чем у сверточной, что положительно сказывается на процессе обучения сети.

Глава 2. Программная реализация

2.1 Построение модели

Для решения данной задачи будем использовать модель капсульной нейронной сети SegCaps [4] (рис. 10).

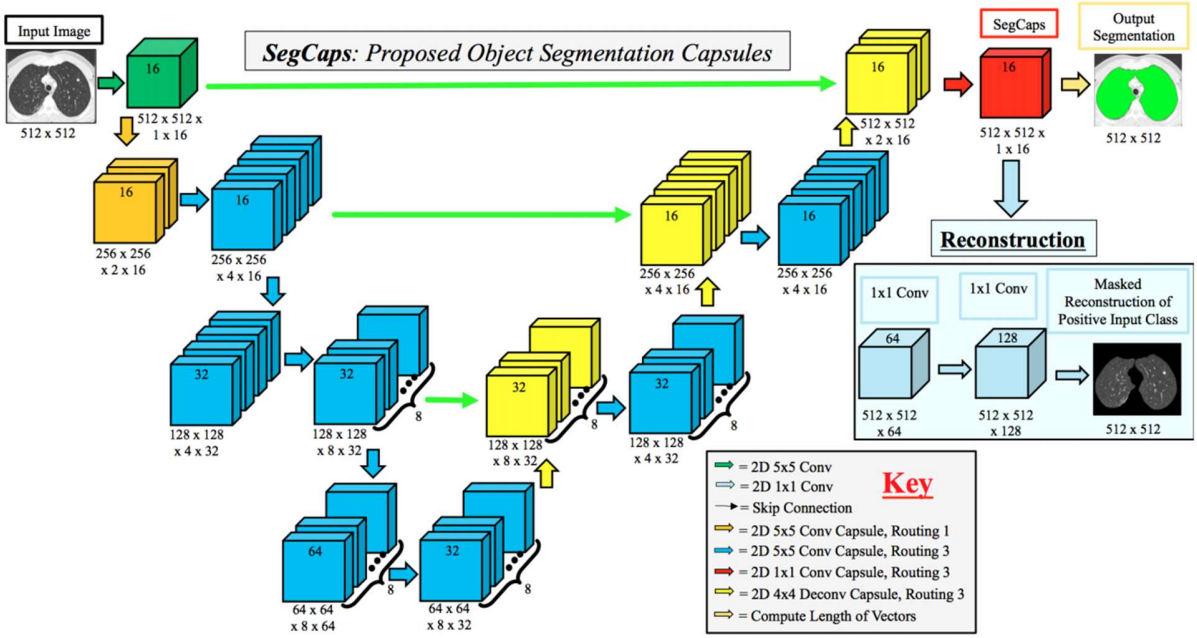

Рис. 10 Архитектура модели SegCaps.

Адаптируем данную модель под задачу предсказания карт глубин. Изменим размерность входного слоя на 480x640 (размер изображений в датасете). Функцию активации последнего слоя поменяем с сигмоидальной на ReLU, так как выходные значения сети должны быть в диапазоне от 0 до 10. В качестве функции потерь будем использовать:

$$L = \frac{1}{n}\sum \frac{(D' - D^*)^2}{D^*},$$

где n – количество элементов в матрице, D^* - истинная карта глубин, D' - предсказанная карта глубин.

Обучение будем производить на платформе Google Colab с использованием видеокарты Nvidia Tesla P100-PCIe 16 GB. В качестве языка программирования

выбран Python 3.6. Для построения и обучения нейронных сетей взята библиотека Keras.

2.2 Результаты

Для обучения модели датасет был разделен на тренировочную, валидационную и тестовую выборки в соотношении 70/20/10. Количество эпох обучения – 100. В качестве метрик использовались среднеквадратичная ошибка (RMSE), среднеквадратичная логарифмическая ошибка ($RMSE_{log}$) и логарифмическая ошибка (log_{10}).

Результаты обучения модели представлены на рисунке 11, а также в таблице 1. В таблице 1 рассматриваются модели, которые обучались на датасете NYU Depth Dataset V2.

Таблица 1 Значение ошибок на тестовой выборке.

Модель	RMSE	$RMSE_{log}$	log_{10}
FCRN [6]	0.581	0.207	0.072
PlaneNet [13]	0.514	0.180	0.060
SegCaps	0.672	0.252	0.098

Процесс обучения занял 10 часов (6 минут на эпоху). Среднее время обработки одного изображения на тестовой выборке составило 1.76 секунды.

Исходя из этих результатов, можно сделать вывод, что предложенная капсульная сеть немного уступает сверточным сетям. Дальнейшая модернизация и усложнение сети может улучшить этот результаты. Также этому может поспособствовать другой датасет с большим количеством изображений.

Рис. 11 Результаты предсказания карты глубин. Сверху вниз: исходное изображение, истинная карта глубины, предсказанная карта глубины.

Выводы

В данной работе решалась задача построения карты глубин по одному изображению с применением капсульных нейронных сетей. Для решения задачи была модернизирована и обучена модель SegCaps, изначально предназначенная для семантической сегментации изображений. Результаты показали, что рассмотренная модель немного уступает сверточным нейронным сетям, но является перспективной для решения задачи построения карты глубин.

Заключение

В ходе выполнения выпускной работы получены следующие результаты, которые выносятся на защиту:

- Предложена модификация модели капсульной нейронной сети SegCaps для построения карты глубин.
- Реализована программа для обучения модели.
- Проведены тесты и сравнения с другими работами.

Список литературы

1. Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, Jiaya Jia Pyramid Scene Parsing Network // CVPR 2017.
2. Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, Jian Sun. Unified Perceptual Parsing for Scene Understanding // Accepted to European Conference on Computer Vision (ECCV) 2018.
3. Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, Jingdong Wang High-Resolution Representations for Labeling Pixels and Regions // CVPR 2019.
4. Rodney LaLonde, Ulas Bagci Capsules for Object Segmentation // 1st Conference on Medical Imaging with Deep Learning (MIDL 2018).
5. Olaf Ronneberger, Philipp Fischer, and Thomas Brox U-net: Convolutional networks for biomedical image segmentation // In International Conference on Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015.
6. Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, Nassir Navab Deeper Depth Prediction with Fully Convolutional Residual Networks // CVPR 2016.
7. Tinghui Zhou, Matthew Brown, Noah Snavely, David Lowe Unsupervised Learning of Depth and Ego-Motion from Video // CVPR 2017.
8. Clément Godard, Oisin Mac Aodha, Gabriel J. Brostow Unsupervised Monocular Depth Estimation with Left-Right Consistency // CVPR 2017.
9. Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, Andreas Geiger Sparsity Invariant CNNs // International Conference on 3D Visio 2017.
10. Sunil Prakash, Gaelan Gu Simultaneous Localization And Mapping with depth Prediction using Capsule Networks for UAVs // CVPR 2018.

11. Sara Sabour, Nicholas Frosst, Geoffrey E Hinton Dynamic Routing Between Capsules // CVPR 2017.
12. Nathan Silberman, Derek Hoiem, Pushmeet Kohli and Rob Fergus Indoor Segmentation and Support Inference from RGBD Images // ECCV 2012.
13. Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, Yasutaka Furukawa PlaneNet: Piece-wise Planar Reconstruction from a Single RGB Image // CVPR 2018