МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования   
**«Национальный исследовательский   
Нижегородский государственный университет им. Н.И. Лобачевского»**

**(ННГУ)**

**Институт информационных технологий, математики и механики**

**Кафедра: математического обеспечения и суперкомпьютерных технологий**

Направление подготовки: «Фундаментальная информатика и информационные технологии»

**ОТЧЕТ**

по учебной практике

**Тема:**

**«Сегментация медицинских изображений.  
Разработка алгоритмов автоматической сегментации данных трехмерного сканирования сердца (КТ, МРТ, УЗИ) с помощью глубокого обучения на языке Python с применением TensorFlow \ PyTorch.»**

**Выполнил:** студент группы 381906-3

**\_\_\_\_\_\_\_\_\_**Олюнин Александр Владимирович

Подпись

**Научный руководитель:**

младший научный сотрудник **\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Васильев Евгений Павлович

Подпись

Нижний Новгород

2021

**Оглавление**

[1. Введение 3](#_Toc93326627)

[2. Постановка задачи 5](#_Toc93326628)

[3. Описание фреймворков TensorFlow и Keras 6](#_Toc93326629)

[3.1 Возможности загрузки dataset 6](#_Toc93326630)

[3.2 Возможности Keras в построении моделей 8](#_Toc93326631)

[3.3 Популярные модели сегментации 11](#_Toc93326632)

[3.4 Расширенные возможности оптимизации процесса обучения 14](#_Toc93326633)

[4. Предварительная обработка изображений 18](#_Toc93326634)

[5. Медицинская часть U-Net 20](#_Toc93326635)

[6. Заключение 24](#_Toc93326636)

[7. Список литературы и источников информации 25](#_Toc93326637)

# 1. Введение

Часто возникают задачи, в которых требуется разбить некоторое изображение на несколько частей с выделением групп, классов по набору отличительных признаков. В современной науке за этот процесс отвечает термин сегментация и находит свое применение в ряде задач, например, таких как поиск и распознавание объектов, нахождение границ и редактирование изображений. Подобные задачи стояли и в практической работе при изучении нейронных сетей. Тем не менее, сам процесс сегментации представляет собой разбиение изображения на области, которые должны обладать одинаковыми признаками. Такое решение находит свое применение, в том числе в медицине, где необходимо выделить участки изображения со здоровыми или пораженными органами, тканями и т.п.

С развитием технологий закономерно появилось несколько способов сегментировать изображения, причем как черно-белые, (особенно полутоновое изображение), так и цветные изображения. В медицине же изображения, как правило, подразумеваются в виде полутона с глубиной цвета от 16 бит, так как меньшего количества диапазона возможных значений яркости не всегда может быть достаточно.

Нужно отметить, что работа по сегментации можно разделить на два следующих вида:

* Интерактивная
* Автоматическая

Отличие заключается, в том, что при интерактивной сегментации требуется ручной труд человек, в то время как при автоматической этот труд можно заменить алгоритмами, в чем и состоит ее главное преимущество. Основное внимание в данной работе будет уделено видам автоматической сегментации изображений. Среди всего разнообразия данных методов, особенно можно выделить:

* Пороговая сегментация
* Наращивание областей
* Выделение границ
* Текстурные методы

Все эти методы не далеко уходят от стандартных решений при обработке изображения, пройденных на курсах компьютерной графики и обработки изображения. Самый простой из всех приведенных выше способов – пороговая сегментация. Суть работы заключается в том, чтобы выделить однородные по интенсивности участки с использованием некоторого порога, определяемого автоматически специальным алгоритмом. В случае применения метода наращивания областей, участки близкие по яркости будут присоединяться в один большой существенный. Выделение границ же подразумевает непосредственное выделение контуров изображения, подобные задачи решает множество фильтров, классическим примером может быть матричный фильтр оператор Собеля. Суть работы таких алгоритмов сводится, в грубом смысле, поиска производной, градиента изображения, выражаясь более мягко, поиск перепадов значений пикселей, что соответствует значениям максимума и минимума кривой. Текстурные методы по большей части применимы к цветным изображениям, как следует из названия, они ориентируются на диффузные свойства поверхности анализируемого объекта. Суть методов заключается в приведении изучаемых изображений к задаче различения уровней яркости, применяя специальные операторы, для этой цели.

*Оператор Собеля*

В дальнейшем речь пойдет о технологиях глубокого обучения в целях сегментации медицинских изображений, с полна реализующих преимущество заменить ручной труд человека на автоматическую работу нейронной сети.

# 2. Постановка задачи

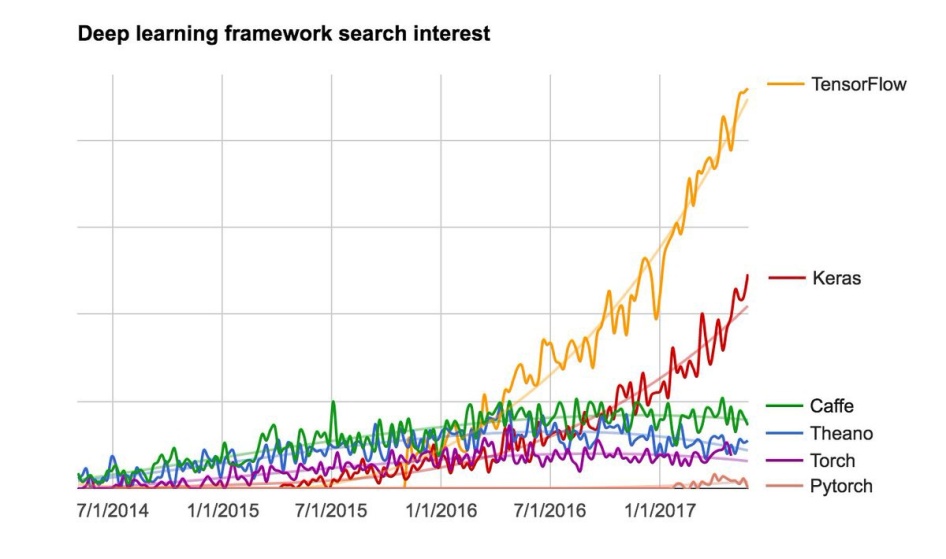
В данной практической работе стоит задача изучить язык программирования Python, основное внимание уделить широким возможностям библиотек, написанным для ускорения вычислений. Познакомиться с технологией глубокого обучения, исследовать возможности библиотек TensorFlow и Keras, разработанных на Python для удобного конструирования и обучения нейронных сетей. Создать и обучить свою собственную нейронную сеть. Изучить основные виды нейронных сетей, использующихся в области глубокого обучения, некоторые способы предобработки изображений при использовании сверточных нейронных сетей. Рассмотреть структуру и работу сети U-Net, созданной для сегментации медицинских изображений. Проанализировать полученные результаты работы и сделать вывод.

# 3. Описание фреймворков TensorFlow и Keras

## **3.1 Возможности загрузки dataset**

Фреймворк Keras разработан для поддержки глубокого обучения для языка Python, при этом является более компактной и удобной некоторой надстройкой над набравшей высокую популярность за последнее время по сравнению с другими библиотекой TensorFlow (рис. 3.1), что благоприятствует быстрому созданию и анализу нейронных сетей. Основными характеристиками можно считать:

* Поддержка переносимости или возможность запустить код на GPU и CPU без изменений
* Дружественный API, упрощающий разработку прототипов моделей глубокого обучения
* Встроенная поддержка основных видов нейронных сетей: прямая, сверточная, рекуррентная.



*Рис. 3.1 Рост количества поисковых запросов для разных фреймворков*

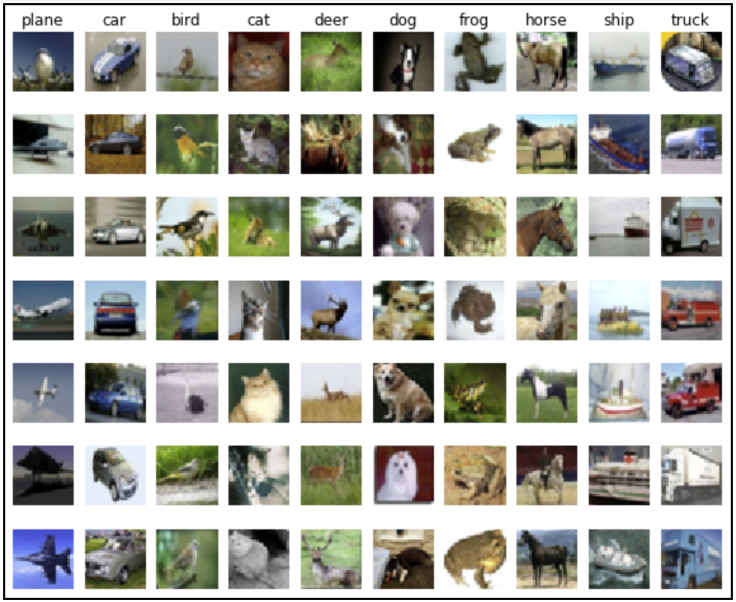
*глубокого обучения по данным Google*

Помимо вышеперечисленного, библиотека Keras обладает еще достаточно большим рядом преимуществ. В данном разделе речь пойдет о встроенных datasets прямо в Keras, и о том, как можно использовать для обучения нейронной сети сторонний dataset.

Итак, для работы с нейронными сетями в Keras предусмотрена поддержка некоторых полезных наборов данных, таких как:

* CIFAR10
* CIFAR100
* IMDB
* Reuters
* MNIST
* Boston\_housing

Наборы данных CIFAR содержат по 50 тыс. каждый учебных изображений размера 32 на 32 пикселя, помеченных в 10 и 100 категориях соответственно.



*Рис. 3.2*

Dataset IMDB содержит 25 тыс. отзывов о фильмах с платформы IMDB, разделенными бинарными метками по их настроению, в виде прямой кодировки самых популярных в отзывах слов (каждое слово кодируется уникальным целым положительным числом) распределенных по убыванию их популярности.

Так же есть такой набор данных, как классифицированные темы новостных лент на 46 видов. Всего в базе содержится более 11 тыс. новостных лент журнала Reuters. Представление данных такое же, как и в IMDB – прямое кодирование проводов.

Самый классический набор данных – это MNIST, который содержит 60 тыс. полутоновых изображений 28 на 28 пикселей, помеченных по виду одной из десяти уникальных арабских цифр на изображении. В dataset в том числе включен набор тестовых изображений, не содержащий меток.

Дополнительный набор данных – boston\_housing, включающий в себя 13 атрибутов, описывающих некоторое состояние рынка цен на дома в пригороде Бостона в конце 70-ых годов прошлого века. Источником данных является библиотека StatLib, представленная в университете в Карнеги-Меллон.

Указанные стандартные datasets можно загрузить используя небольшую конструкцию программы на языке Python

from keras.datasets import. <name of dataset>

(x\_train, y\_train),(x\_test, y\_test) = <name of dataset>.load\_data()

Кроме того, данные наборы можно загрузить сразу в более удобном виде, для этого есть инструменты использующиеся как аргументы функции load\_data(). Например, наиболее применимыми из них возможность загрузить данные в перемешанном виде, с усечением последовательности, с разделением для резервирования данных в качестве тестового набора.

В практической работе, были рассмотрены и использованы все виды встроенных datasets. Примеры кода будут располагаться здесь (ссылка на github).

Помимо встроенных datasets, Keras располагает возможностями использовать независимые от него наборы данных. В практической работе было рассмотрено два способа:

1. Использование директории, в которой располагается большое множество фотографий.

2. Таблица excel, заполненная данными.

В обоих случаях данные были взяты с открытого источника сайта для соревнований в области глубокого обучения Kaggle. Для работы с данными также была необходима библиотека Python для работы с проводником – path. Примеры кода будут располагаться здесь (ссылка на github).

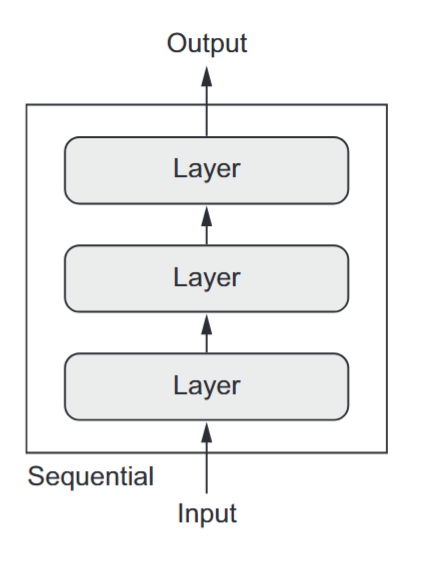
## **3.2 Возможности Keras в построении моделей**

Существуют три основных семейства архитектур моделей нейронных сетей, поддерживаемых в Keras:

* Полносвязные
* Сверточные
* Рекуррентные

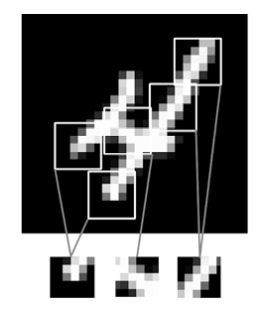
Данные типы вбирают в себя некоторые важные характеристики, позволяющие эффективно применять их в различных задачах, таких как бинарная классификация, множественная классификация, однозначная или многозначная, задача регрессии (предсказание значения по предыдущим показаниям). Решение каждой задачи есть построение пространства гипотез, в котором производится поиск наилучшей подходящей модели предсказателе. Выбор архитектуры полностью зависит от вида стоящей задачи.

Например, при работе с векторными данными наилучшим выбором будет использовать полносвязную нейронную сеть. Данная сеть представляет собой набор слоев Dense, которые строят свои выводы исходя не из структуры входных данных, а из их наличия. То есть, их работа заключается в поиске отношения между двумя любыми, присутствующими во входных данных элементах. Они проводят качественный анализ данных и часто становятся неотъемлемой частью в построении более сложных моделей, используются в качестве верхних слоев такой модели. На практике они применялись почти в каждом примере, и чисто их использование было рассмотрено для решения каждой из вышеперечисленных задач. Тут стоит выделить специальные настройки под ту или иную задачу. Для бинарной классификации нужно использовать функцию потерь binary\_crossentropy и функцию активации sigmoid. Вторая позволяет достичь на выходе распределение вероятности от 0 до 1 принадлежности к определенному классу. Для однозначной классификации правильно использовать функцию потерь categorical\_crossentropy и функцию активации softmax. В случае многозначной классификации функция categorical\_crossentropy заменяется на binary\_crossentropy. Для задачи регрессии следует использовать функцию потерь средней квадратичной ошибки или средней абсолютной ошибки, mae и mse соответственно. Функция потерь при этом не требуется.



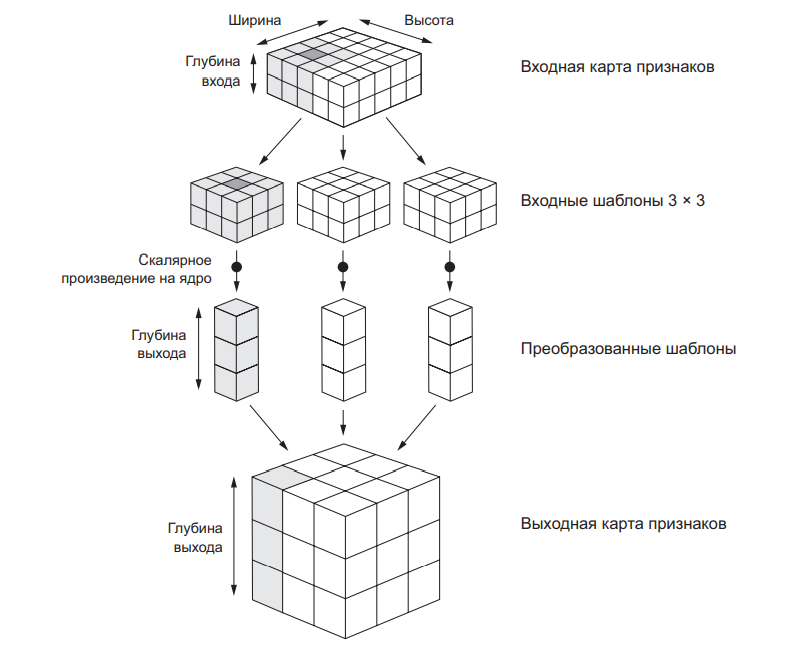
*Рис. 3.3. Схематический вид полносвязной нейронной нейронной сети*

Для работы с многомерными тензорами данных наилучшим выбором будет использование многомерных сверточных сетей. Они представляют собой несколько последовательно выстроенных слоев Conv (свертка) и MaxPooling любой из доступных размерностей (1D, 2D, 3D). В отличие от слоев полносвязных сетей, они проводят оценку локальных шаблонов, сочетаний значений элементов, что позволяет найти выделенный шаблон в любом месте данных. Такое свойство называется инвариантностью относительно переноса. Таким образом, они позволяют сделать эффективную пространственную выжимку информации из входных данных и, как было сказано ранее, далее полученные шаблоны передаются в слои полносвязной сети для дальнейшей классификации.



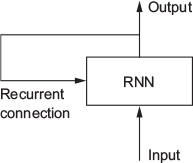
*Рис. 3.4 Разбиение изображения на локальные шаблоны*

Принцип работы свертки заключается в проходе некоторой многомерного вектора по тензору данных и применения операции скалярного произведения к соответствующим ей элементам этих данных. В случае двумерного тензора (часто просто изображения) эта операция соответствует матричному произведению. В компьютерной графике такой двумерный вектор имел название ядро, в глубоком обучении имеет название фильтра. Таких фильтров в слое сверточной сети может быть не мало, часто степени двойки, на выходе получается некоторое множество изображений с выделенными на них признаками изображения. Эти изображения называются картами признаками. Также, естественно, что при использовании в операции свертки матрицы размерности более одного, то карта признаков будет уменьшаться, во избежание этого процесса есть специальные методы дополнения границ исходного изображения, но в целом, это используется на часто, так как суть сверточных сетей выделить главную информацию, для чего, и предназначена следующая операция – MaxPooling. Данный слой производит выбор максимальных значений по соседям, что снижает размерность данных, что необходимо для сохранения размеров карты признаков в разумных пределах с ростом числа признаков, и позволяет последующим сверточным слоям анализировать пространство входных признаков на большем протяжении.



*Рис. 3.5 Принцип работы свертки*

Последним из рассматриваемых является класс рекуррентных нейронных сетей. Наиболее эффективно их использовать при обработке последовательности данных, основываясь на их существенной зависимости от параметра времени. В таком случае, использование сверточных сетей уже не является эффективным, так как данные не обладают свойством инвариантности относительно переноса. В Keras они представлены слоями SimpleRNN, GRU, LSTM. На выходе можно иметь либо один вывод результирующей последовательности, либо несколько последовательностей от каждого из слоев. На практике были применены в ходе предсказания температуры на следующие дни по метерологическим данным.



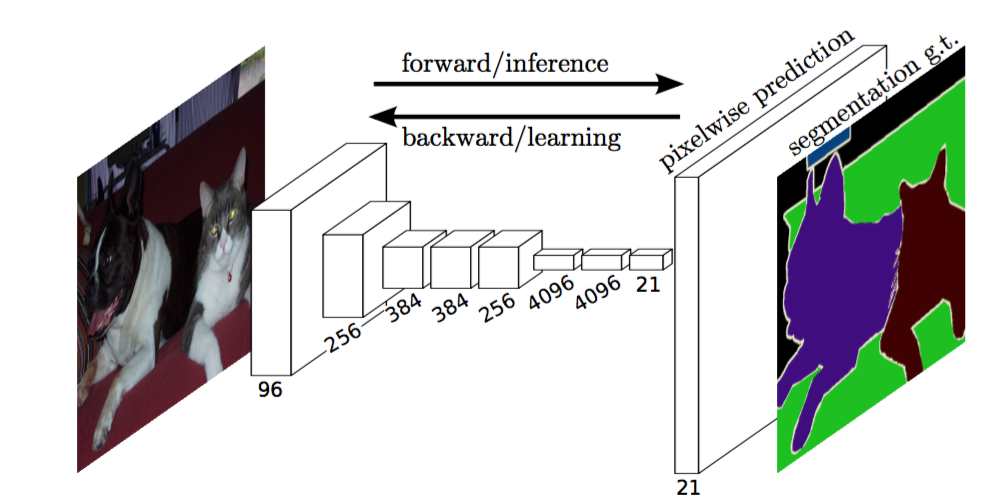
*Рис. 3.6 Принцип работы рекуррентной нейронной сети*

## **3.3 Популярные модели сегментации**

Обычно модели сегментации, основанные на глубоком обучении, строятся на основе некоторой предварительно обученной на наборе ImageNet нейронной сети. В таком случае применяется заморозка сверточной основы, а верхние слои классификатора, сегментирования начинают свое обучение сначала – со случайных весов. Такую модель можно выбрать из уже заключенных в Keras предварительно обученных сверточных сетей, таких как ResNet, модель от Microsoft, имеет большое количество слоев и оставшихся соединений, VGG-16, более легкая сеть, так как меньше количество слоев, использовалась на практике, смотрите код на github, MobileNet, модель от Google, наиболее оптимизирована для работы на устройствах с ограниченными ресурсами, тех же мобильных телефонах.

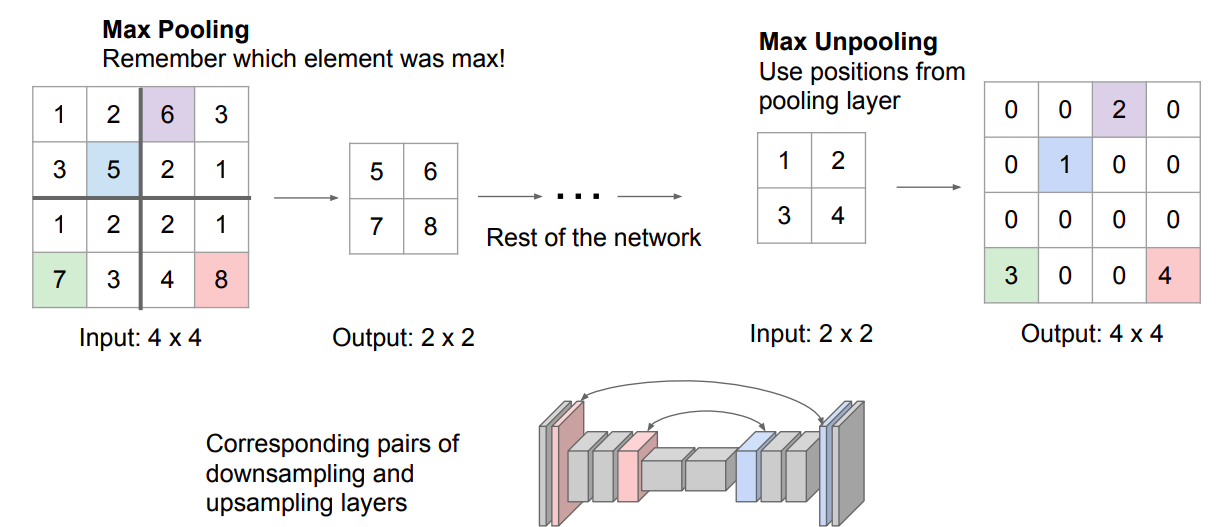
В Keras возможна реализация следующих моделей сегментации

1. FCN ­ полностью сверточная нейронная сеть, самая простая и популярная модель семантической сегментации. Суть работы заключается в уменьшении входного изображения до меньшего размера, тем самым получая увеличенный вектор пространства каналов, при помощи серии сверток. Такой набор сверток обычно именуется кодером. Затем, выход работы кодера подвергается дискретизации при помощи билинейной интерполяции или серии транспонированных сверток, который называется декодером. Существенным минусом в работе транспонированной свертки является появление артефактов шахматной доски из-за неравномерного перекрытия выходных данных. Еще одним недостатком служит разрешение на границах из-за потери информации в процессе кодирования.



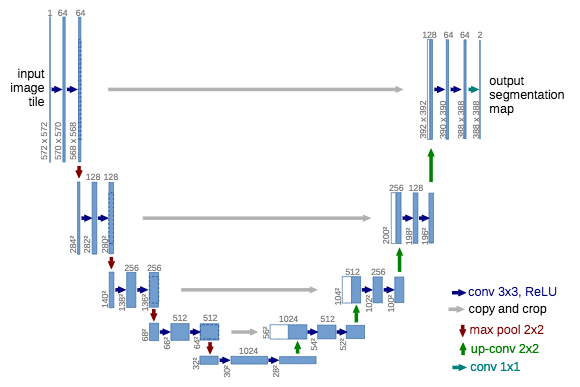
*Рис. 3.7 Архитектура FCN*

2. SegNet – основана на идее кодера и декодера, как и FCN, однако, оптимизация состоит в том, что их уровни симметричны друг другу. Операция дискретизации уровня декодера использует максимальный индекс пула соответствующего ему уровня кодера. В отличие от U-Net еще не использует пропуск соединений. Для повышения частоты дискретизации нет обучаемых параметров.



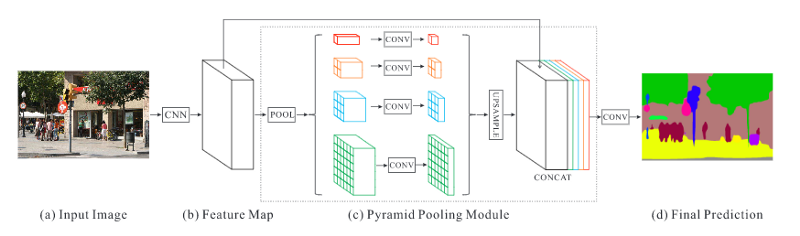
*Рис. 3.8 Архитектура SegNet*

3. U-Net – использует идею FCN, однако также предусматривает решение тех проблем декодировщика, которые были отмечены выше. Предполагает наличие симметричной обработки данных с использованием пропуска соединения с выхода блока свертки в соответствующий ему блок транспонированной свертки на том же уровне. Добавление пропуска соединения позволяет лучше представлять информацию из нескольких масштабов изображения. Информация с верхних уровней помогает модели лучше классифицировать участки изображения, в то время как с мелких уровней она помогает сегменту лучше локализироваться.



*Рис. 3.8 Архитектура U-Net*

4. PSPNet – сеть пирамидального распознавания сцены сочетает в себе 4 свертки разного масштаба пирамиды. Предварительно обученная модель базового слоя и применение свертки извлекают карту признаков из входного изображения, имеющую размер 1/8 этого изображения. Далее происходит согласованное объединения всех уровней свертки и повышенной выборки. В конце полученную структуру необходимо еще раз прогнать через сверточный слой.



*Рис. 3.9 Архитектура PSPNet*

## **3.4 Расширенные возможности оптимизации процесса обучения**

В Keras существуют несколько инструментов, способных улучшить процесс обучения моделей. Они позволяют существенно сократить время разработки хорошей нейронной сети и тем самым повысить ее конечное качество. Разберем несколько таких оптимизирующих механизмов, представленных внутри Keras.

1. Callback – объект класса, который обладает всей информацией о состоянии обучения нейронной сети и при этом может влиять на него. Данный механизм позволяет экономить время и ресурсы, в случае однозначно неудачной архитектуры модели, сохранять удачные попытки подбора весов и следить за обучением на более качественном уровне. Вот, некоторые возможности, которые он дает:

* Зафиксировать состояние модели на определенных этапах обучения, т.е. сохранение весов в процессе обучения
* Приостановить обучение модели в случае череды неудач - длительного роста значений функции потерь
* Динамической корректировки некоторых гиперпараметров обучения, таких как шаг обучения оптимизатора
* Журналирования оценок функции потерь на проверочных данных с целью дальнейшего изучения протекания процесса обучения

Причем уникальность в том, что callback можно создавать самому с нужными характеристиками под конкретную задачу. Более того, можно создавать сразу несколько в виде списка объектов класса callback, например:

callbacks\_list = [keras.callbacks.EarlyStopping(

                    monitor = "val\_acc",

                    patience = 1),

                    keras.callbacks.ModelCheckpoint(

                    monitor = "val\_loss",

                    filepath = "my\_model.h",

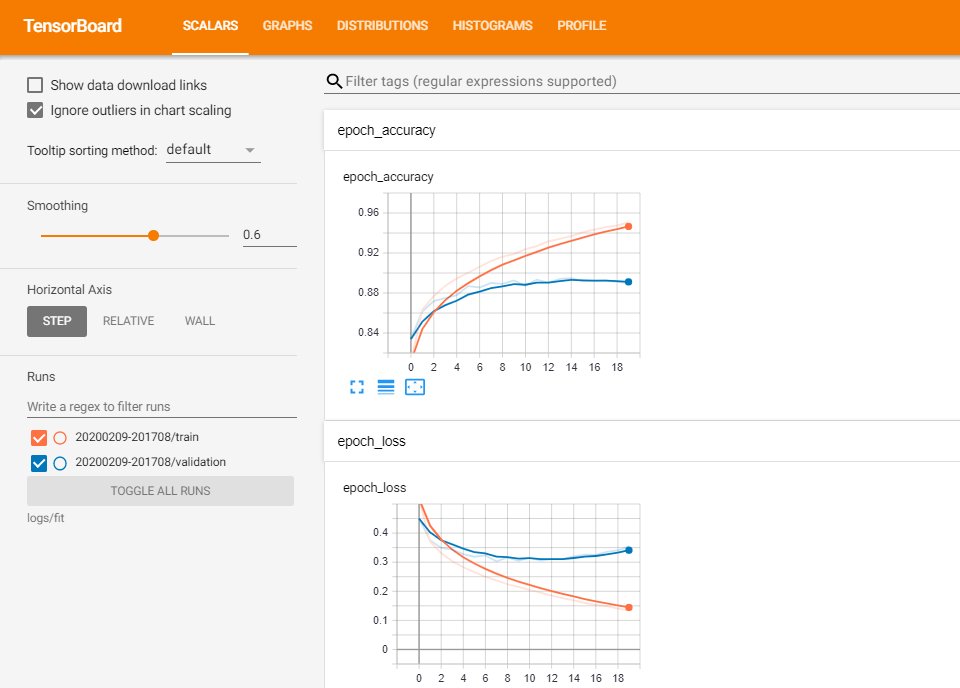
                    save\_best\_only = True)]

В данном коде EarlyStopping обратный вызов, который прервет обучение в случае, если метрики не улучшатся в течение 2 эпох (параметр patience). Второй обратный вызов в списке ModelCheckpoint сохраняет метрики в ходе обучения, причем только в том случае, если метрики улучшились (параметр save\_best\_only). В случае необходимости создать свой callback, достаточно просто создать свой класс, унаследованный от стандартного класса Callback.

2. Главная цель экпериментов с обучением нейронной сети – получение информации о качестве работы модели, с этой задачей может помочь справиться полезный инструмент TensorBoard. Его работа поддерживается внутренним браузером фреймфорка TensorFlow. Данный механизм обеспечивает поддержкой следующих возможностей:

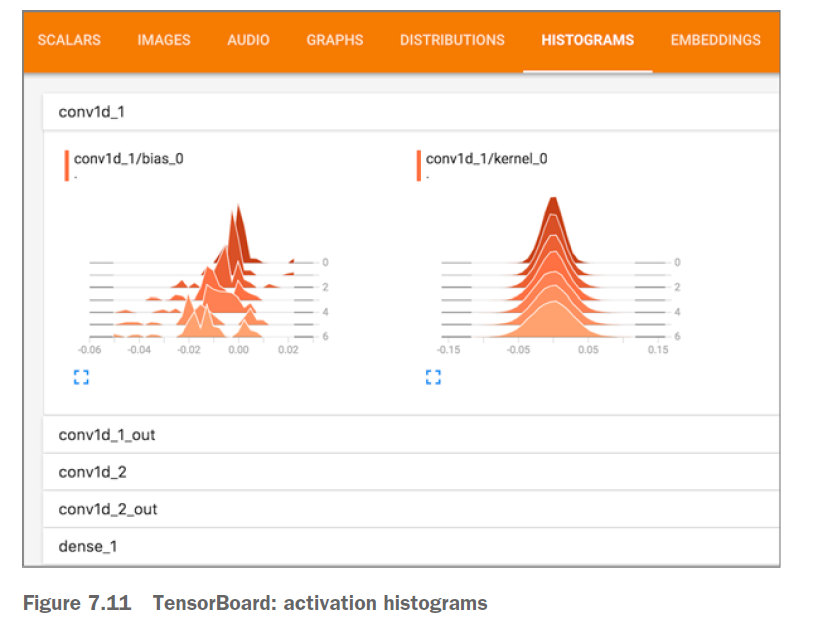
* Мониторинг метрик в ходе обучения
* Визуализация архитектуры модели
* Вывод гистограмм активаций и градиентов
* Исследование векторных представлений в трехмерном пространстве.

Использовать данный механизм можно в виде обратного вызова. Далее, передать его в функцию обучения модели. Расположение вывода работы будет располагаться на локальном адресе, соответственно достаточно воспользоваться браузером. Главная страница TensorBoard демонстрирует динамически обновляющиеся графики метрик, получаемых в ходе обучения и проверки на проверочных данных.



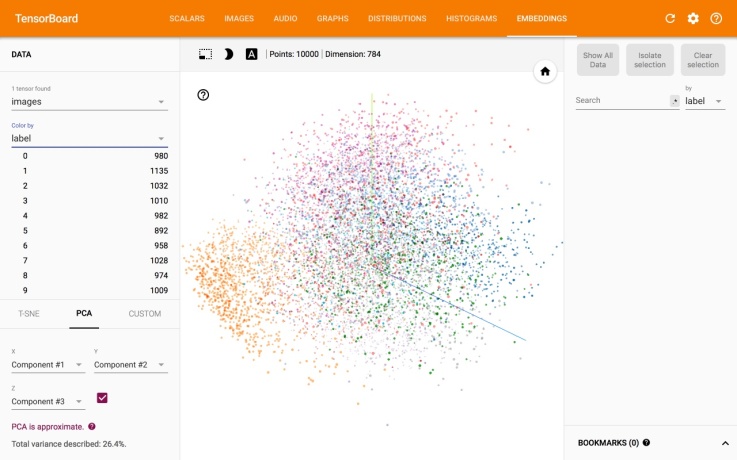
*Рис. Мониторинг метрик*

Помимо этого, на вкладке Histograms располагаются гистограммы активаций.



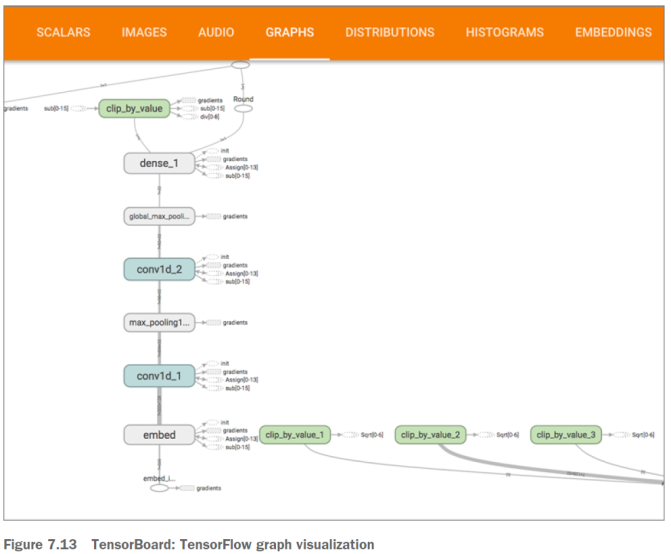
*Рис. Гистограммы активаций*

В случае работы с векторным представлением слов, можно получить красивый график пространственного отношения всех слов во входном словаре. Удобство в том, что TensorBoard способен автоматически понизить размерность пространства для удобства работы с помощью алгоритма снижения размерности по выбору программиста.



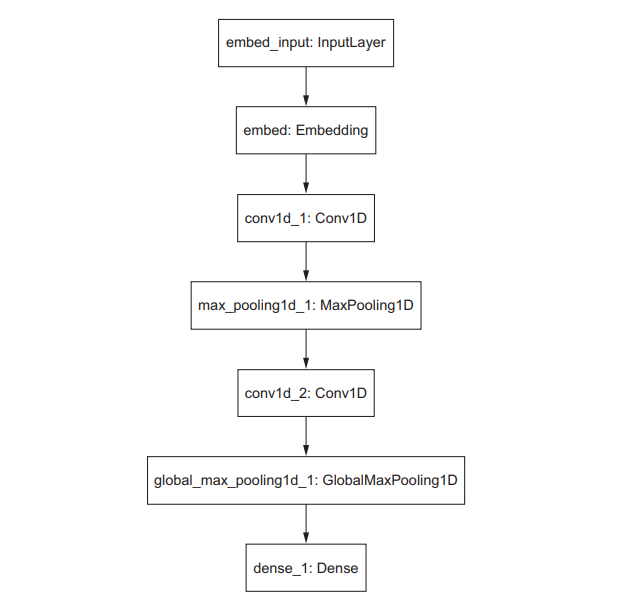
*Рис. Интерактивная карта трехмерного простраства векторных представлений слов*

На вкладке Graphs можно наблюдать всю полноту низкоуровневого построения модели, которая происходит в TensorFlow при создании модели.



*Рис. Граф операций TensorFlow*

3. Подобные представления о структуре нейронной сети можно получить другим механизмом, который поддерживает Keras – утилиты keras.utils.plot\_model, она позволяет получить изображение графа слоев модели. Пример того, как это будет выглядеть представлен на рис.

**

*Рис. Модель в виде графа слоев*

Таким образом, фреймворк Keras содержит несколько удобных инструментов для работы с обучением нейронных сетей, способных вывести процесс обучения и проверки на качественно новый уровень.

# 4. Предварительная обработка изображений

Процедура предварительной обработки используется практически всегда после получения информации с датчика и заключается в операции усреднения и выравнивания гистограмм, применении различного типа фильтров для исключения помех, возникающих в результате аппаратной дискретизации и квантования и для подавления внешних шумов. Изображения, подлежащие такой обработке можно разделить на 3 типа:

* Многоуровневые и бинарные изображения
* Бинарные изображения или битовые карты изображений
* Изображения характерных линий контуров или скелетов объектов на изображении

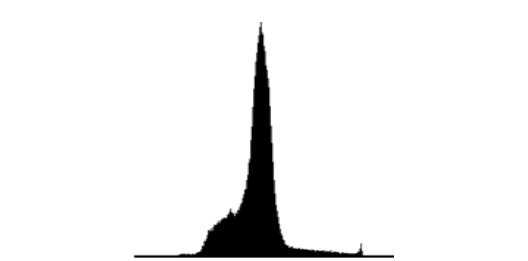
Самым распространенным и сложным для анализа является первый тип, в то время как два других идут на встречу упрощения этого процесса.

С одной стороны задача подавления помех входит в задачу улучшения изображения, а с другой – ее можно рассматривать как часть задачи сегментации. Так же как необходимо отделить шум от изображения, так и в задаче сегментации необходимо отделить объекты от области фона. Данные операции являются трудоемкими, в связи с тем, что понятия «объект», «шум», «фон», «граница объекта» условны. Методы и алгоритмы сегментации можно рассматривать, как формализацию понятия выделяемости объекта из фона.

Влияние предварительной обработки может существенно зависеть от способа сегментации изображения. Например, часто для улучшения работы метода пороговой сегментации изображения применяют инвертирование изображения. Более того, данная предобработка полезна для использования в непосредственно в глобальной обработке, причиной тому способствует тот факт, что операция инвертирования помимо инвертирования цвета, в том числе инвертирует интенсивность, что делает светлые участки изображения темнее, а темные наоборот – светлее. Это в свою очередь в лучшую сторону влияет на работу алгоритма глобальной пороговой обработки. Из курса компьютерной графики известна формула, по которой можно осуществить процедуру инвертирования:

где - значение яркости пикселя изображения с координатами , а – новое значение яркости этого пикселя.

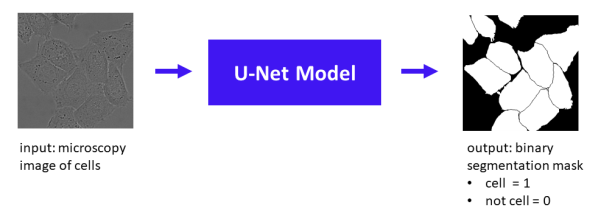
Однако, процедура инвертирования не всегда улучшает работу данного алгоритма, в том случае, если гистограмма изображения представляет собой очень сложную форму (рис. 4.1), то тогда инвертирование не имеет смысла, поскольку сложность гистограммы не изменится и корректность работы фильтра будет оставлять желать лучшего. Более того, как показывает практика, применение этой предобработки для других методов сегментации не дает сильного улучшения работы алгоритма.



*Рис. 4.1 Пример сложной гистограммы*

# 5. Медицинская часть U-Net

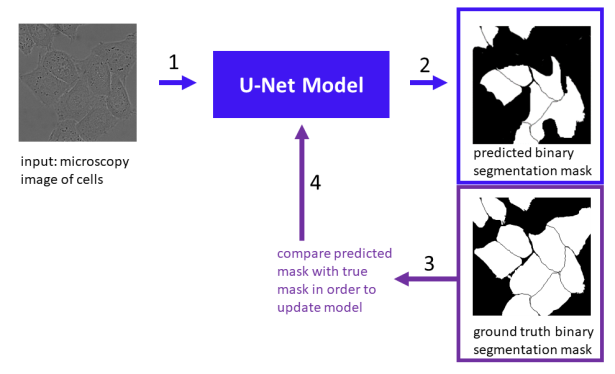
Оригинальная статья U-Net, набравшая огромную популярность за последние 6 лет, описывает архитектуру модели семантической сегментации. Причем, изначально она была описана в контексте биомедицинских изображений, но впоследствии развития была также успешно применена и к другим видам изображений. Ключевая идея работы модели U-Net заключается в получении бинарной маски сегментации (рис. 5.1).



*Рис. 5.1 Принцип работы U-Net архитектуры*

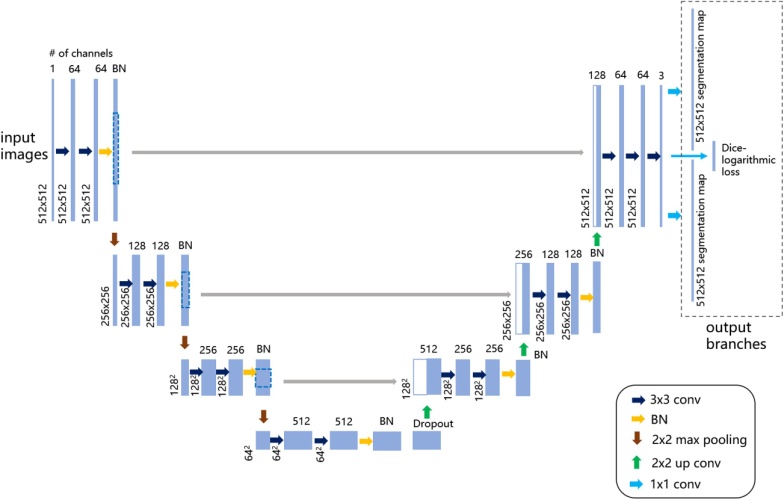
В данном примере входным изображением является полутоновое изображение клеток, сделанное с помощью микроскопа. Бинарная маска, которая получается на выходе, кодирует 0 как элементы фона изображения, в том числе и границы между клетками, а 1 как сами клетки. Отметим, что данный пример иллюстрирует семантическую сегментацию, так как выделенных участков всего два: передний план и задний. В типичной задаче сегментации есть необходимость разделить передний план на еще несколько классов.

Обучение модели заключается в том, чтобы сравнивая выходную бинарную маску предсказания и истинную бинарную маску добиваться уменьшения значений функции потерь, за счет изменения весов механизмом обратного распространения ошибки.



*Рис. 5.2 Схема обучения U-Net модели*

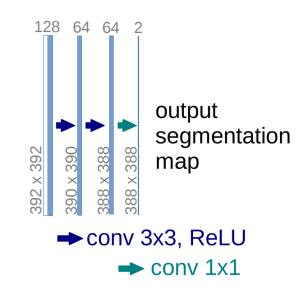
На рис. 5.3 можно наблюдать саму архитектуру U-Net, что и демонстрирует название, она имеет форму в виде буквы U. Сначала U-Net получает низкое разрешение исходного изображения, сжимая его при помощи классической сверточной нейронной сети. Затем, увеличивает образец, чтобы получить на выходе сегментированное изображение. В правом нижнем углу изображения есть описание, какому типу операций соотносится каждая стрелка в архитектуре.

**

*Рис. 5.3 Архитектура U-Net*

Первая часть архитектуры или кодер, связан со второй частью или декодером серыми стрелками, которые обозначают операцию копирования. Ее работа заключается в том, что выходная карта признаков кодировщика каждого уровня копируется и соединяется (операция конкатинации) с картой признаков соответствующего ему уровню декодировщика, что помогает нейронной сети обучаться сегментации высокого разрешения.

Рассмотрим работу последнего уровня декодера. В самом конце сети данные проходят через слой свертки 1 на 1 пикселей с отображением 64 каналов вектора признаков в необходимое количество классов. В данном примере рассматривается два класса, поэтому последний слой будет выглядеть так:

**

*Рис. 5.4*

Для получения семантической сегментации изображения используется функция активации pixel-wise softmax:

где ­ значения активаций в канале признаков *k* пикселя с координатой *x* и *K –* количество нужных классов.

В качестве функции потерь используется cross-entropy loss, которая имеет вид:

*w* – весовые коэффициенты, - активации последнего слоя, т.е. предсказание модели.

Весовые коэффициенты вычисляются автоматически с использованием полученных ранее истинных сегментированных изображений при помощи стандартных методов компьютерного зрения. В данном примере дополнительно применяется математическая морфология для выявления тонких границ между клетками, которые сильно увеличивают весовые коэффициенты. Данное новшество позволяет изменять веса модели таким образом, чтобы эти границы действительно присутствовали и были нарисованы нужных местах. В общем, весовые коэффициенты в функции потерь служат цели выявления границ между клетками.

К сожалению, построение и обучение такой модели сегментации может быть очень время затратным, так как присутствует необходимость практически вручную нарисовать истинное сегментированное изображение. К тому же, набор таких данных может быть ограниченным. Для того, чтобы справиться с этой проблемой можно воспользоваться методами расширения тренировочных данных. Он состоит в том, чтобы применить случайные деформации к изображению, такие как сдвиги пикселей, вращения изображения, изменения оттенков серого или другие эластичные преобразования. Эластичные преобразования наиболее эффективны в случае медицинских изображений, потому что они устойчивы к такого вида деформации, и все еще выглядят реалистично после этих изменений.

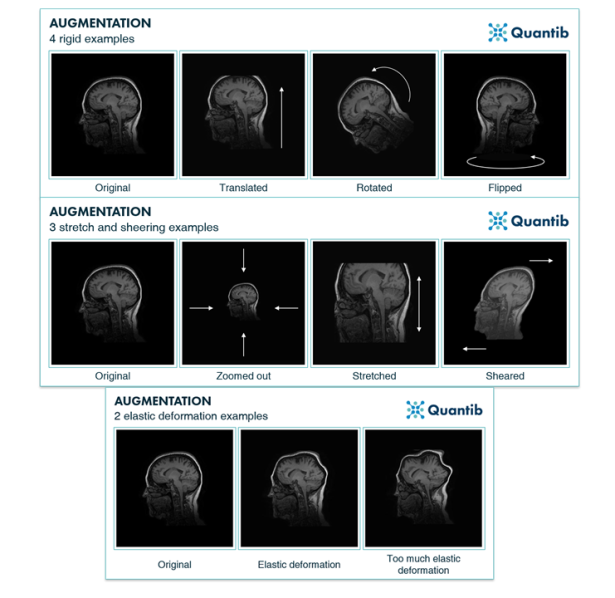


Рис. 5.5 Эластичные деформации

Таким образом, модель U-Net достигла больших успехов не только в области медицинских изображений разного рода, но и в других видах изображений.

# 6. Заключение

**Вывод:** в данной практической работе были изучены различные методы сегментирования изображения, как стандартных автоматических, так и с использованием технологий глубокого обучения. Был изучен язык Python, применены на практике библиотеки, расширяющие его возможности в вычислениях и построении нейронных сетей. Более того, было написаны реализации полносвязных, рекуррентных и сверточных нейронных сетей разной мерности для решения различного рода задач. Были рассмотрены специальные возможности фреймворков Keras и TensorFlow, повышающие качество и скорость экспериментов в целях построения наиболее удачных моделей нейронных сетей. Проанализированы наиболее распространенные решения для семантической и обычной сегментации изображений. В качестве такой сегментации медицинских изображений наиболее приемлемым вариантом может служить архитектура U-Net.

# 7. Список литературы и источников информации

* Шолле Франсуа. Глубокое обучение на Python. – СПб.: Питер, 2018 – 400 с.
* Национальная библиотека им. Баумана. URL: <https://ru.bmstu.wiki/Текстурные_методы_(Сегментация)>
* Малышев В. Д. Разработка онлайн сервиса для сегментации медицинских изображений. Минск, 2019
* Бойко Д. О. Филатова А.Е. Обзор методов сегментации медицинских изображений.
* Divam Gupta. A Beginner's guide to Deep Learning based Semantic Segmentation using Keras. URL: <https://divamgupta.com/image-segmentation/2019/06/06/deep-learning-semantic-segmentation-keras.html>
* Distill. Deconvolution and Checkerboard Artifacts. URL: <https://distill.pub/2016/deconv-checkerboard/>
* Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical. Image Segmentation. URL: <https://arxiv.org/pdf/1505.04597.pdf>
* Digital Image Processing (CS/ECE 545) Lecture 10: Discrete Fourier Transform (DFT)
* ProgrammerSought. The most comprehensive semantic segmentation review in history (FCN, UNet, SegNet, Deeplab, ASPP...) URL: <https://programmersought.com/article/45131571697/>
* А. Ю. Тропченко, А.А. Тропченко Методы вторичной обработки и распознавания изображений, Санкт-Перебург, 2015 URL: <https://books.ifmo.ru/file/pdf/1798.pdf>
* [Jae Duk Seo](https://medium.com/@SeoJaeDuk?source=post_page-----d07231eb29bf-----------------------------------). Medical Image Segmentation [Part 1] — UNet: Convolutional Networks with Interactive Code. URL: <https://medium.com/@SeoJaeDuk/medical-image-segmentation-part-1-unet-convolutional-networks-with-interactive-code-d07231eb29bf>
* Glass Box. Segmentation: U-Net, Mask R-CNN, and Medical Applications. URL: <https://glassboxmedicine.com/2020/01/21/segmentation-u-net-mask-r-cnn-and-medical-applications/>